UTS:ENGINEERING AND INFORMATION TECHNOLOGY

UTS UNIVERSITY OF TECHNOLOGY, SYDNEY

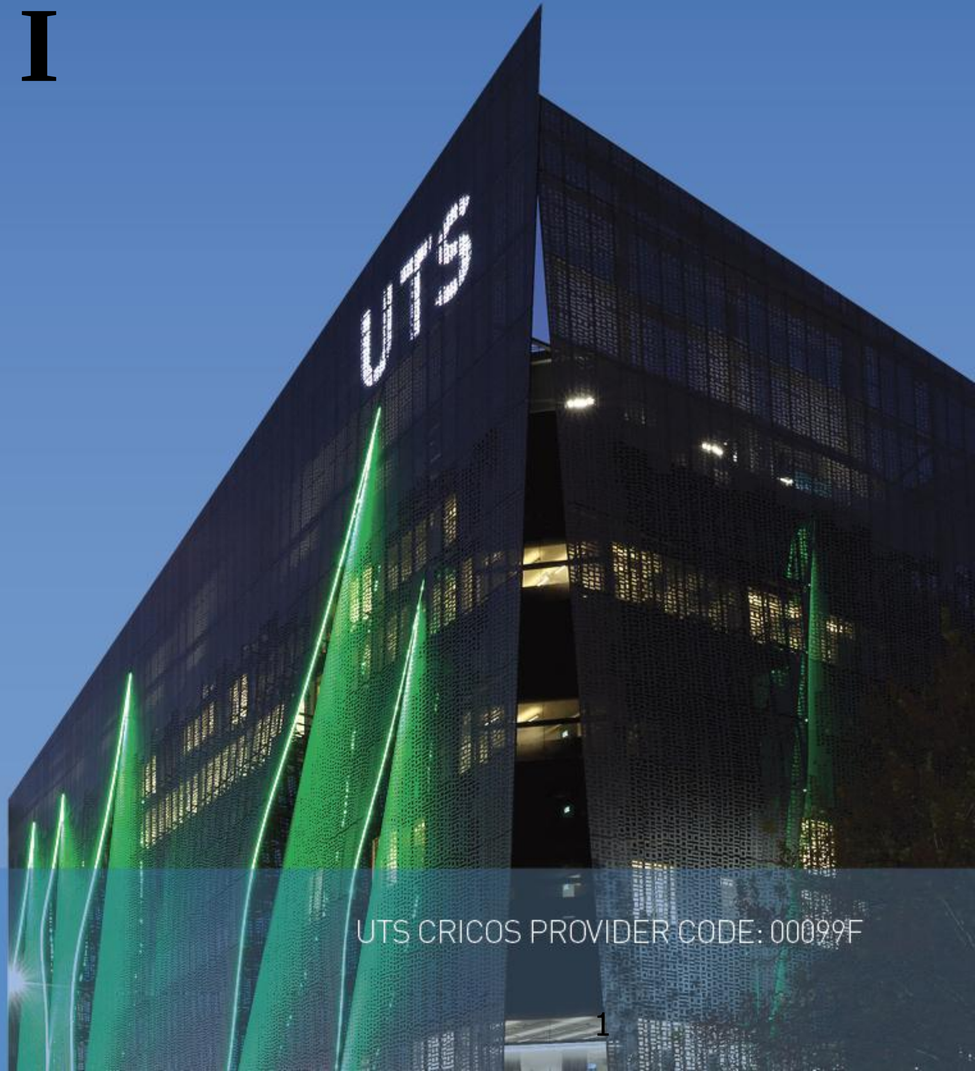
lecture 7: SQL I
Simple Query

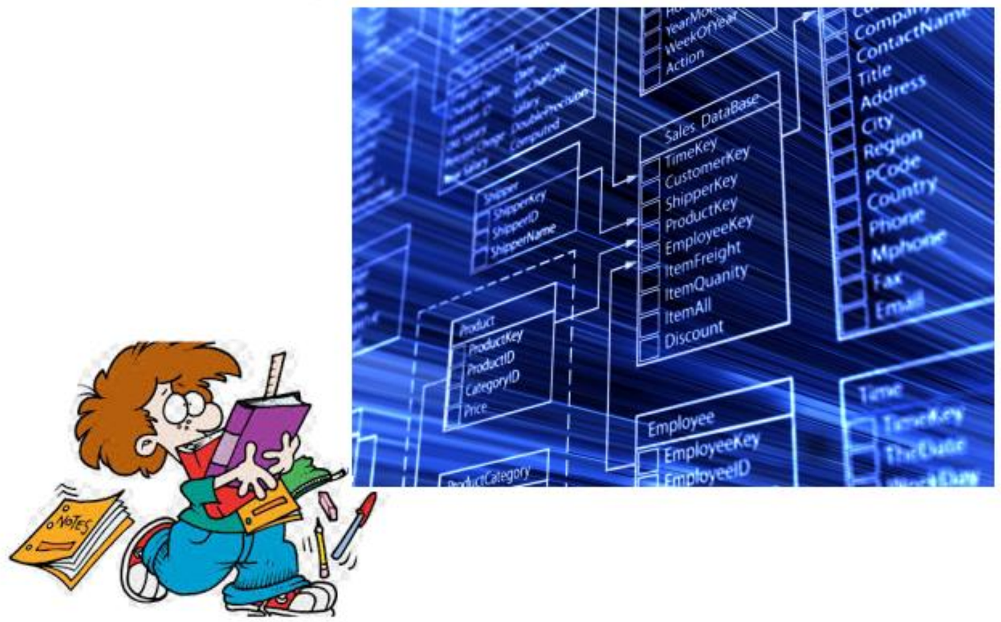
Main reference:

Modern Database Management, 11th Edition
Chapter 6: Introduction to SQL

Subject Coordinator and Instructor:

Dr. Danna (Fahimeh) Ramezani





Select * from MyLife_T where COVID19 is gone;

MyLife_T

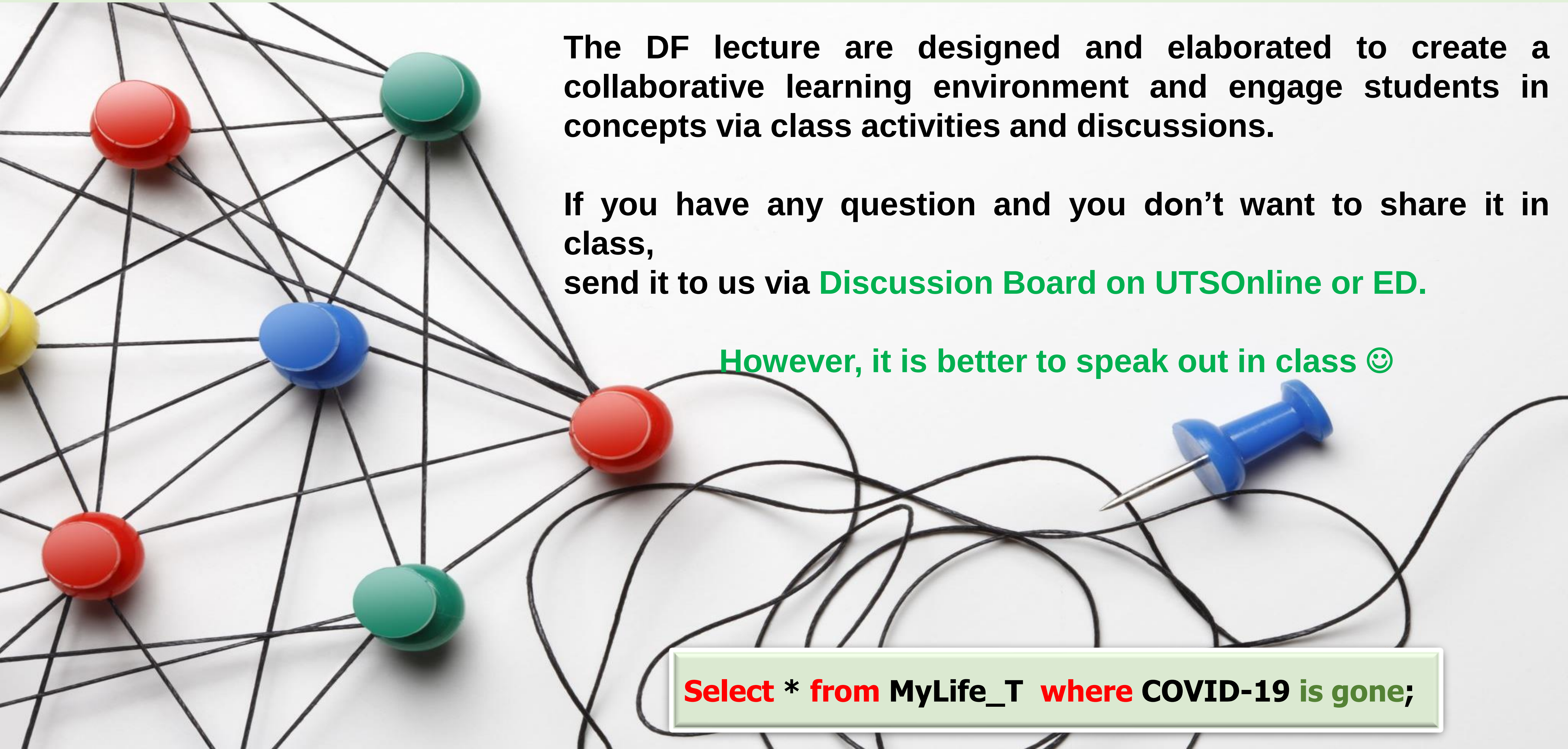
HappinessID	HappinessName	HppinessStartDate	HppinessEndDate	COVID_19
1755	Pass DF	09/03/2020	null	Gone
1899	Graduated	09/03/2019	null	Came
...	...	...	...	...

MySuccess_T

SuccessID	SuccessName	SuccessDate	HappinessID
1967	Got HD Grade in PF	8/10/2019	1755
2055	Got HD Grade in DF	null	1755
3798	Start my job in NASA	null	1899
...	...	...	...

Select * from MyLife_T where Covid_19='Gone';

Participations and Discussions



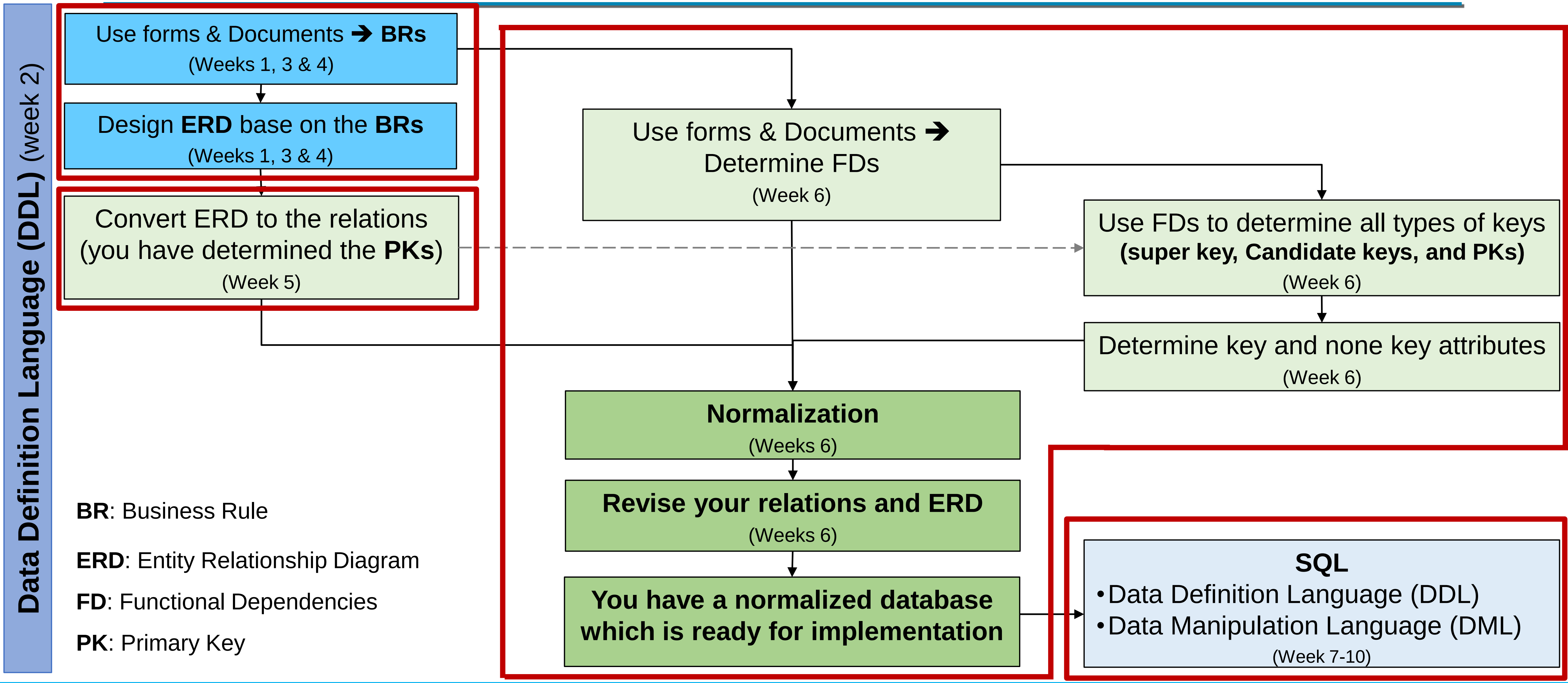
The DF lecture are designed and elaborated to create a collaborative learning environment and engage students in concepts via class activities and discussions.

If you have any question and you don't want to share it in class,
send it to us via Discussion Board on UTSOnline or ED.

However, it is better to speak out in class ☺

Select * from MyLife_T where COVID-19 is gone;

Subject Flowchart


Data Definition Language (DDL) (week 2)
Use forms & Documents → BRs (Weeks 1, 3 & 4)
Design ERD base on the BRs (Weeks 1, 3 & 4)
Convert ERD to the relations (you have determined the PKs) (Week 5)
Use forms & Documents → Determine FDs (Week 6)
Use FDs to determine all types of keys (super key, Candidate keys, and PKs) (Week 6)
Determine key and none key attributes (Week 6)
Normalization (Weeks 6)
Revise your relations and ERD (Weeks 6)
You have a normalized database which is ready for implementation
SQL
•Data Definition Language (DDL)
•Data Manipulation Language (DML)
(Week 7-10)

BR: Business Rule

ERD: Entity Relationship Diagram

FD: Functional Dependencies

PK: Primary Key

Subject Overview

- **Design Entity Relationship Diagram (ERD)**
  - **Week 1: Data Modelling I (Conceptual Level):** Entity, Attributes, PK, FK, …
  - **Week 2: Data Definition Language (DDL):** Create tables, constraints, insert, …
  - **Week 3: Data Modelling II (Conceptual Level):** Associative, Weak, …
  - **Week 4: Data Modelling III (Conceptual Level):** Subtype/Supertype
  - **Week 5: Convert ERD to Relations (Logical Level)**
  - **Week 6: Functional Dependencies, and Normalization**
- **Data manipulation**
  - **Week 7: Simple Query**
  - **Week 8: Multiple Table Queries**
  - **Week 9: Subquery**
  - **Week 10: Correlated Subquery**

Lecture Seven Objectives:

3. Simple query

3.1. SELECT Statement: Select & From Clauses

3.2. SELECT Statement: Where Clause

Operators: >, >=, …, Like, Between, not null, (), NOT, and, or, etc.

3.3. SELECT Statement: Order By Clauses

3.4. SELECT Statement: Group By Clauses

Aggregate Functions …

3.5. SELECT Statement: Having

3.6. SQL statement processing order

3.7. Views

3.1. SELECT Statement

- Select statement is used for queries on single or multiple tables

  Select column1, column2 From Table1 Where [Condition on rows]

  Group by column1, column2 Having [Condition on groups]

  Order by column1

- Clauses of the SELECT statement:
  - SELECT: List the columns (and expressions) to be returned from the query
  - FROM: Indicate the table(s) or view(s) from which data will be obtained
  - WHERE: Indicate the conditions under which a row will be included in the result
  - GROUP BY: Indicate categorization of results
  - HAVING: Indicate the conditions under which a category (group) will be included
  - ORDER BY: Sorts the result according to specified criteria

3.1. SELECT Statement: Select & From Clauses

Clauses of the SELECT statement:

- SELECT
  - List the columns (and expressions) to be returned from the query
- FROM
  - Indicate the table(s) or view(s) from which data will be obtained
- WHERE
  - Indicate the conditions under which a row will be included in the result
- GROUP BY
  - Indicate categorization of results
- HAVING
  - Indicate the conditions under which a category (group) will be included
- ORDER BY
  - Sorts the result according to specified criteria

3.1.1.The Simplest Query

- List all the data in a Product_T table

```
Select * from product_T ;
```

All SQL commands

end with a semicolon “ ; ”

Note: SQL keywords (e.g. “select”, “from”) are NOT case sensitive (other things can be case sensitive).

```
 productid | productlineid |    productdescription     | productfinish | productstandardprice | productonhand
-----------+---------------+---------------------------+---------------+----------------------+---------------
         1 |             1 | Cherry End Table          | Cherry        |               175.00 |             0
         2 |             1 | Birch Coffee Tables       | Birch         |               200.00 |             0
         3 |             1 | Oak Computer Desk         | Oak           |               750.00 |             0
         4 |             1 | Entertainment Center      | Cherry        |              1650.00 |             0
         5 |             2 | Writer's Desk             | Oak           |               325.00 |             0
         6 |             1 | 8-Drawer Dresser          | Birch         |               750.00 |             0
         7 |             3 | 48 Bookcase               | Walnut        |               150.00 |             0
         8 |             3 | 48 Bookcase               | Oak           |               175.00 |             0
         9 |             3 | 96 Bookcase               | Walnut        |               225.00 |             0
        10 |             3 | 96 Bookcase               | Oak           |               200.00 |             0
        11 |             1 | 4-Drawer Dresser          | Oak           |               500.00 |             0
        12 |             1 | 8-Drawer Dresser          | Oak           |               800.00 |             0
        13 |             1 | Nightstand                | Cherry        |               150.00 |             0
        14 |             2 | Writer's Desk             | Birch         |               300.00 |             0
        17 |             3 | High Back Leather Chair   | Leather       |               362.00 |             0
        18 |             4 | 6' Grandfather Clock      | Oak           |               890.00 |             0
        19 |             4 | 7' Grandfather Clock      | Oak           |              1100.00 |             0
        20 |             2 | Amoire                    | Walnut        |              1200.00 |             0
        21 |             1 | Pine End Table            | Pine          |               256.00 |             0
        24 |             5 |                           |               |                 0.00 |             0
        25 |             2 |                           |               |                 0.00 |             0
(21 rows)
```

3.1.2. Subset of columns

➢ **Just name the columns you want**

Select productdescription, productfinish, productstandardprice from product_t;

```
productdescription     | productfinish | productstandardprice
-------------------------+---------------+----------------------
 Cherry End Table        | Cherry        |               175.00
 Birch Coffee Tables     | Birch         |               200.00
 Oak Computer Desk       | Oak           |               750.00
 Entertainment Center    | Cherry        |              1650.00
 Writer's Desk           | Oak           |               325.00
 8-Drawer Dresser        | Birch         |               750.00
 48 Bookcase             | Walnut        |               150.00
 48 Bookcase             | Oak           |               175.00
 96 Bookcase             | Walnut        |               225.00
 96 Bookcase             | Oak           |               200.00
 4-Drawer Dresser        | Oak           |               500.00
 8-Drawer Dresser        | Oak           |               800.00
 Nightstand              | Cherry        |               150.00
 Writer's Desk           | Birch         |               300.00
 High Back Leather Chair | Leather       |               362.00
 6' Grandfather Clock    | Oak           |               890.00
 7' Grandfather Clock    | Oak           |              1100.00
 Amoire                  | Walnut        |              1200.00
 Pine End Table          | Pine          |               256.00
                         |               |                 0.00
                         |               |                 0.00
```

- **Column names are comma separated.**
- **We can specify any ordering of columns we want.**

3.1.3. Eliminating duplicate rows in result: Distinct

Question: Determine the types of Product finish in product table.

```
Select * from product_T;
```

```
 productid | productlineid |   productdescription    | productfinish | productstandardprice | productonhand
-----------+---------------+-------------------------+---------------+----------------------+---------------
         1 |             1 | Cherry End Table        | Cherry        |               175.00 |             0
         2 |             1 | Birch Coffee Tables     | Birch         |               200.00 |             0
         3 |             1 | Oak Computer Desk       | Oak           |               750.00 |             0
         4 |             1 | Entertainment Center    | Cherry        |              1650.00 |             0
         5 |             2 | Writer's Desk           | Oak           |               325.00 |             0
         6 |             1 | 8-Drawer Dresser        | Birch         |               750.00 |             0
         7 |             3 | 48 Bookcase             | Walnut        |               150.00 |             0
         8 |             3 | 48 Bookcase             | Oak           |               175.00 |             0
         9 |             3 | 96 Bookcase             | Walnut        |               225.00 |             0
        10 |             3 | 96 Bookcase             | Oak           |               200.00 |             0
        11 |             1 | 4-Drawer Dresser        | Oak           |               500.00 |             0
        12 |             1 | 8-Drawer Dresser        | Oak           |               800.00 |             0
        13 |             1 | Nightstand              | Cherry        |               150.00 |             0
        14 |             2 | Writer's Desk           | Birch         |               300.00 |             0
        17 |             3 | High Back Leather Chair | Leather       |               362.00 |             0
        18 |             4 | 6' Grandfather Clock    | Oak           |               890.00 |             0
        19 |             4 | 7' Grandfather Clock    | Oak           |              1100.00 |             0
        20 |             2 | Amoire                  | Walnut        |              1200.00 |             0
        21 |             1 | Pine End Table          | Pine          |               256.00 |             0
        24 |             5 |                         |               |                 0.00 |             0
        25 |             2 |                         |               |                 0.00 |             0
(21 rows)
```

3.1.3. Eliminating duplicate rows in result: Distinct

Question: Determine the types of Product finish in product table.

Select productfinish from product_T;

```
 productfinish
---------------
 Cherry
 Birch
 Oak
 Cherry
 Oak
 Birch
 Walnut
 Oak
 Walnut
 Oak
 Oak
 Oak
 Cherry
 Birch
 Leather
 Oak
 Oak
 Walnut
 Pine

(21 rows)
```

How to eliminates duplicate records from the results?

3.1.3. Eliminating duplicate rows in result: Distinct

```
Select distinct(productfinish) from product_t;
```

```
 productfinish
---------------
 Birch
 Cherry
 Leather
 Oak
 Pine
 Walnut

(6 rows)
```

3.2. SELECT Statement: Where Clause

Clauses of the SELECT statement:

- SELECT
  - List the columns (and expressions) to be returned from the query
- FROM
  - Indicate the table(s) or view(s) from which data will be obtained
- WHERE
  - Indicate the conditions under which a row will be included in the result
- GROUP BY
  - Indicate categorization of results
- HAVING
  - Indicate the conditions under which a category (group) will be included
- ORDER BY
  - Sorts the result according to specified criteria

3.2. Where Clause: Subset of rows: Question

- Run this query first:

```
Select productdescription, productfinish, productstandardprice
from product_t;
```

- Now answer this question:

Question: How can we determine “product description”, “product finish” and “product standard price” just for products with “standard price” more than $275?

3.2.1. Subset of rows: Extract rows you want by using where

Question: How can we determine “product description”, “product finish” and “product standard price” just for products with “standard price” more than $275?

```
Select productdescription, productfinish, productstandardprice
from product_t
where productstandardprice >275;
```

```
   productdescription     | productfinish | productstandardprice
--------------------------+---------------+----------------------
 Oak Computer Desk        | Oak           |               750.00
 Entertainment Center     | Cherry        |              1650.00
 Writer's Desk            | Oak           |               325.00
 8-Drawer Dresser         | Birch         |               750.00
 4-Drawer Dresser         | Oak           |               500.00
 8-Drawer Dresser         | Oak           |               800.00
 Writer's Desk            | Birch         |               300.00
 High Back Leather Chair  | Leather       |               362.00
 6' Grandfather Clock     | Oak           |               890.00
 7' Grandfather Clock     | Oak           |              1100.00
 Amoire                   | Walnut        |              1200.00
(11 rows)
```

3.2.2 SELECT Example Using Alias

- Alias is an alternative column header name

```
Select productdescription as name,
       productstandardprice as price
from product_t
where productfinish = 'Oak';
```

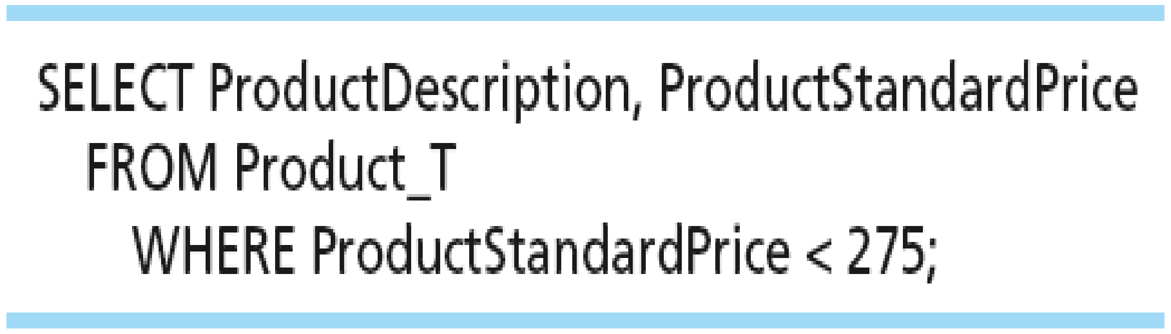
SQL uses single quotes for strings, not double quotes → See 'Oak'

```
       name            | price
-----------------------+---------
 Oak Computer Desk    |  750.00
 Writer's Desk        |  325.00
 48 Bookcase          |  175.00
 96 Bookcase          |  200.00
 4-Drawer Dresser     |  500.00
 8-Drawer Dresser     |  800.00
 6' Grandfather Clock |  890.00
 7' Grandfather Clock | 1100.00
(8 rows)
```

COMPARISON OPERATORS

3.2.3. Comparison Operators That Are Used in WHERE Clause

Question: Find products with standard price less than $275



```
SELECT ProductDescription, ProductStandardPrice
   FROM Product_T
      WHERE ProductStandardPrice < 275;
```

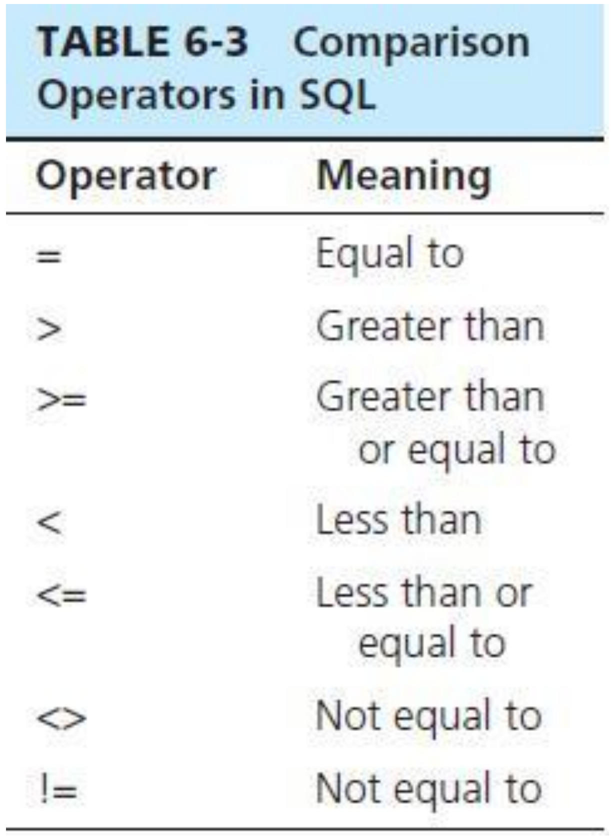


TABLE 6-3 Comparison Operators in SQL

Operator	Meaning
=	Equal to
>	Greater than
>=	Greater than or equal to
<	Less than
<=	Less than or equal to
<>	Not equal to
!=	Not equal to

3.2.4. Other Operators That Are Used in WHERE Clause

- **Operators that are used in where clause:**

1. **Between**
2. **And / Or**
3. **Like**
4. **is not null**
5. **In**

3.2.4.1. Between operator (can be used in WHERE clause)

Question: Determine “product description”, “product finish” and “product standard price” just for products with “standard price” more than $200 and less than $300?

```
Select productdescription, productfinish, productstandardprice
from product_t
where productstandardprice >200 and productstandardprice <300;
```

```
Select productdescription, productfinish, productstandardprice
from product_t
where productstandardprice between 200 and 300;
```

3.2.4.2. AND & OR operators (can be used in WHERE clause)

- The **WHERE** clause can contain several conditions linked by **AND** or **OR**.
- In a **WHERE** containing one or more **ANDs** **all** specified conditions must be true.
- In a **WHERE** containing one or more **ORs**, **at least one** of the conditions must be true.
- If you mix **AND**s and **OR**s, the **AND**s have precedence.

Note: By default, the processing order of Boolean operators is (), then NOT, then AND, then OR

Note: By default, the processing order of Boolean operators is (), then NOT, then AND, then OR

- Operators have the precedence levels: **(), then NOT, then AND, then OR**
- An operator on higher levels is evaluated before an operator on a lower level.
- When two operators in an expression have the same precedence level, they're evaluated **left to right** based on their position in the expression.
- We can use parentheses to override the defined precedence of the operators in an expression.

Note: By default, the processing order of Boolean operators is (), then NOT, then AND, then OR

Question

Question: Determine the value of X in the following statements:

X = (Condition-1) AND (Condition-2)

(Condition-1 is True)

(Condition-2 is True) Then x= True

X = (Condition-1) AND (Condition-2)

(Condition-1 is True)

(Condition-2 is False) Then x= False

X = (Condition-1) OR (Condition-2)

(Condition-1 is True)

(Condition-2 is False) Then x= True

Note: By default, the processing order of Boolean operators is (), then NOT, then AND, then OR

Question: Determine the value of X based on AND and OR priorities

X = (1 > 2) AND (Sun is black) OR (sea is blue) AND (12 < 14)

- ✓ **(1>2) AND (Sun is black) is False**
- ✓ **(sea is blue) AND (12< 14) is True**
- ✓ **x= False OR True**
- ✓ **X is True**

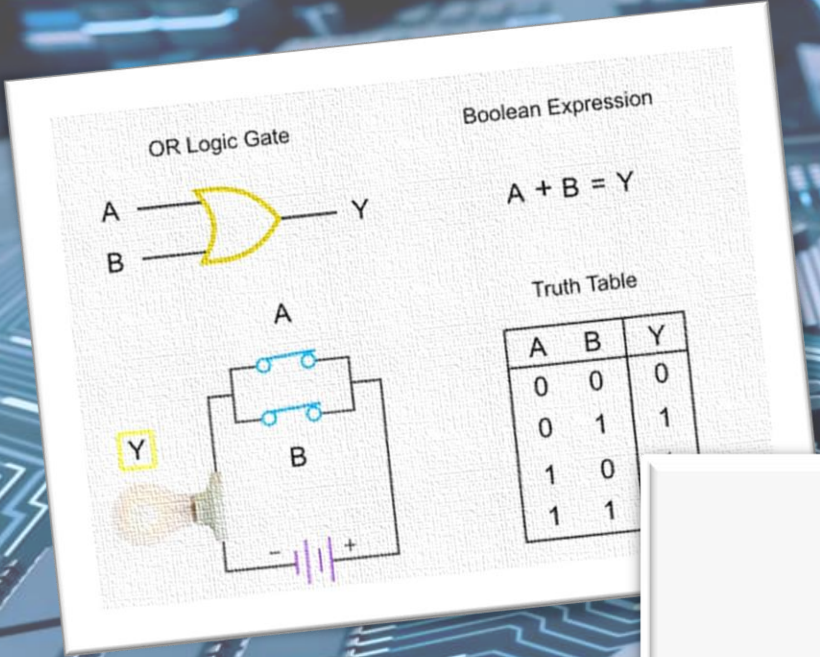
3.2.4.3. Like operator (can be used in WHERE clause)

- The LIKE operator allows you to compare **strings** using **wildcards**.
- For example, the % wildcard in ‘%Table’ indicates that all strings that have any number of characters preceding the word “Table” will be allowed.

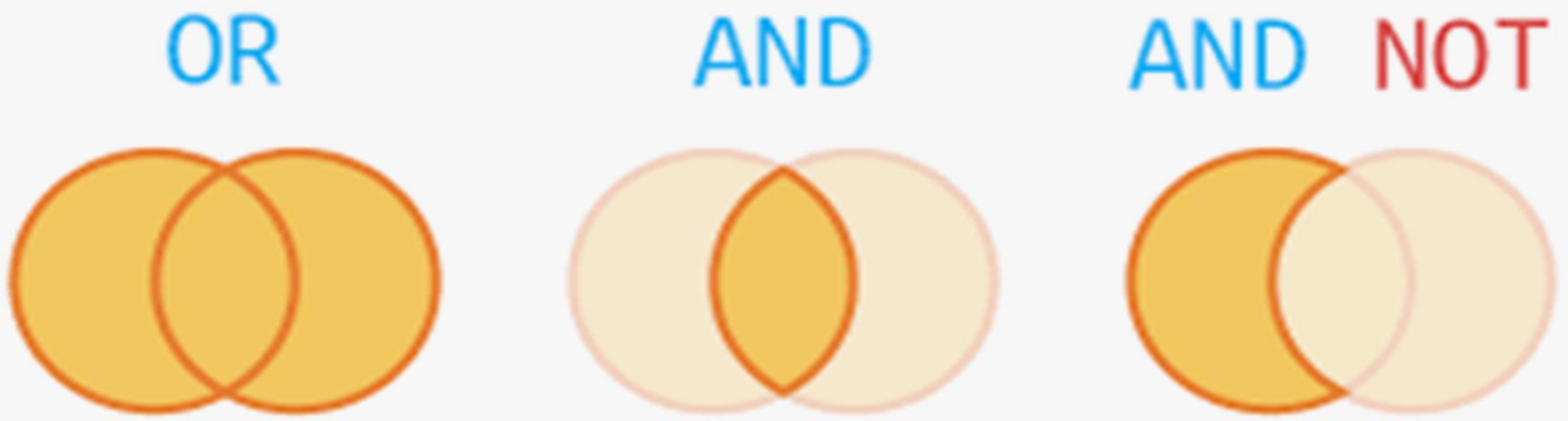
Question: Determine product descriptions that end with word Table.

```
Select productdescription from product_t
Where productdescription Like ‘%Table’;
```

Examples to use AND, OR, NOT, and LIKE Operators



A	B	Y
0	0	0
0	1	1
1	0	
1	1	



Black OR White **Black AND White** **Black AND NOT White**

SELECT Example – Boolean Operators and Like

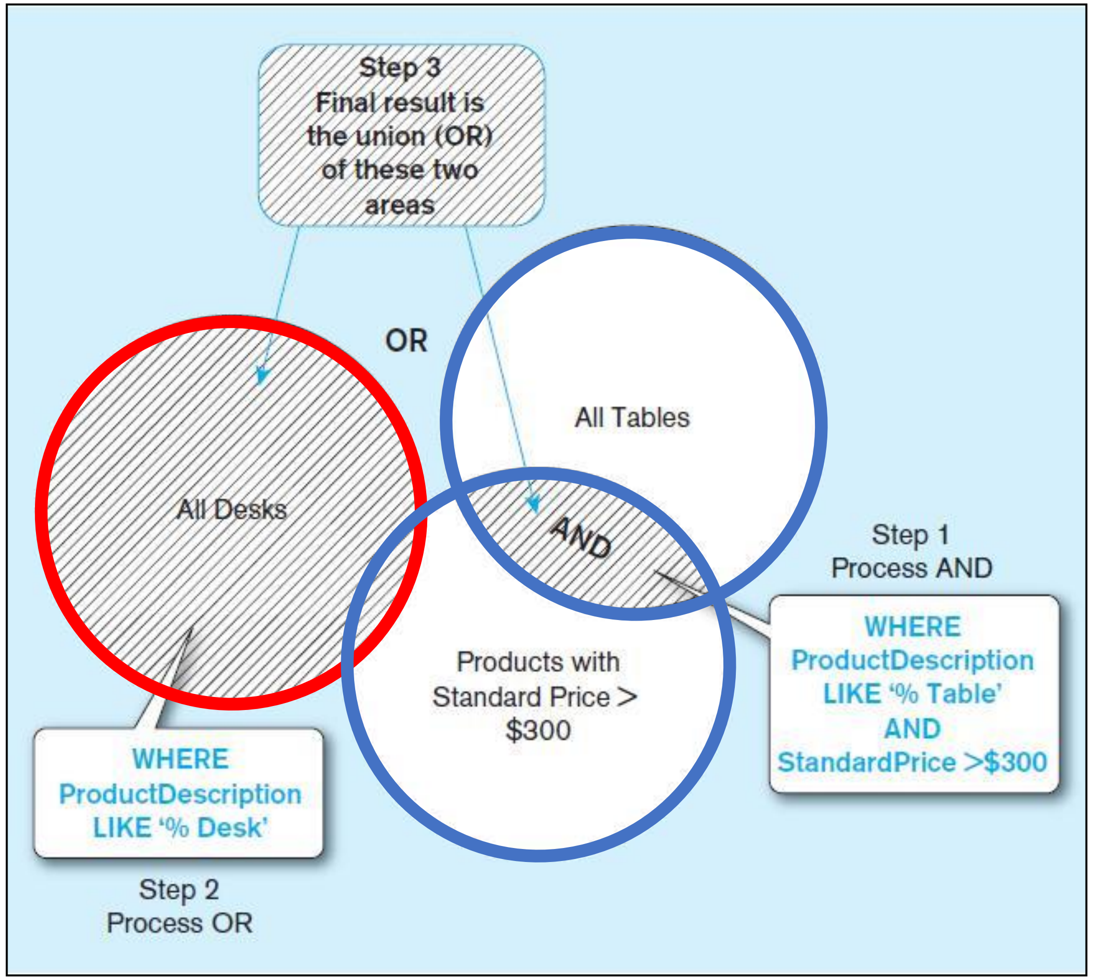
- We use AND, OR, NOT, and Like operators to customize conditions in WHERE clause

Question: Determine information about Desks or Tables with standard price more than $300.

```
Select productdescription, productfinish, productstandardprice
from product_t
where productdescription like '%Desk' OR
        productdescription like '%Table' AND
        productstandardprice > 300;
```

Boolean Query (A): Without Use of Parentheses (Figure 6-8)

```
Select productdescription, productfinish, productstandardprice
from product_t
where productdescription like '%Desk' OR
      productdescription like '%Table' AND
      productstandardprice > 300;
```


Step 3
Final result is the union (OR) of these two areas
OR
All Tables
All Desks
AND
Step 1
Process AND
Products with Standard Price > $300
WHERE ProductDescription LIKE '% Table' AND StandardPrice >$300
WHERE ProductDescription LIKE '% Desk'
Step 2
Process OR

Note: By default, processing order of Boolean operators is

1. () then
2. NOT then
3. AND then
4. OR

SELECT Example – Boolean Operators and Like

```
Select productdescription, productfinish, productstandardprice
from product_t
where productdescription like '%Desk' or
          productdescription like '%Table' and
          productstandardprice > 300;
```

productdescription	productfinish	productstandardprice
Oak Computer Desk	Oak	750.00
Writer's Desk	Oak	325.00
Writer's Desk	Birch	300.00
Pine End Table	Pine	356.00

Note: By default, processing order of Boolean operators is

1. **() then**
2. **NOT then**
3. **AND then**
4. **OR**

SELECT Example – Boolean Operators

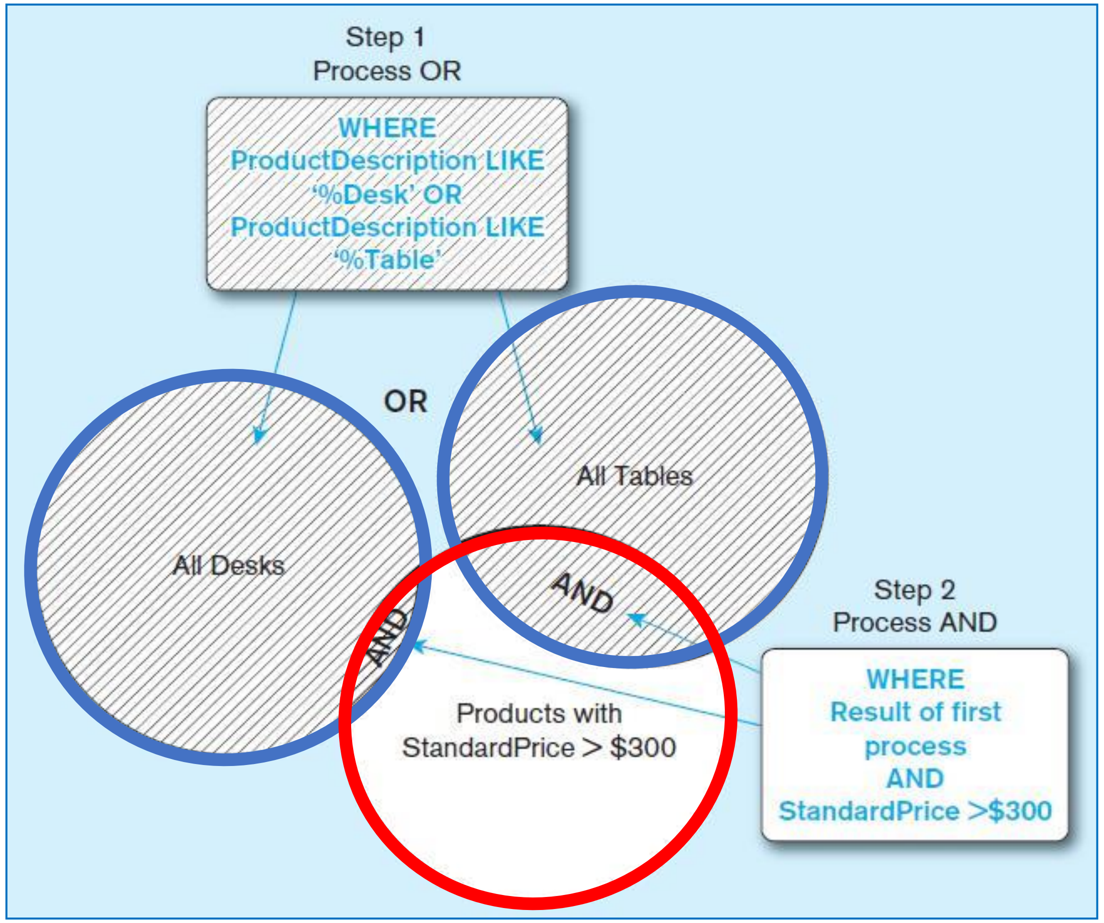
- Parentheses override the normal precedence of Boolean operators.

```
Select productdescription, productfinish, productstandardprice
from product_t
where ( productdescription like '%Desk' OR
     productdescription like '%Table' ) AND
     productstandardprice > 300;
```

With parentheses, you can override normal precedence rules. In this case parentheses make the **OR take place before the AND**.

Boolean Query (B): With Use of Parentheses (Figure 6-9)

```
Select productdescription, productfinish, productstandardprice
from product_t
where ( productdescription like '%Desk' OR
        productdescription like '%Table' ) AND
        productstandardprice > 300;
```



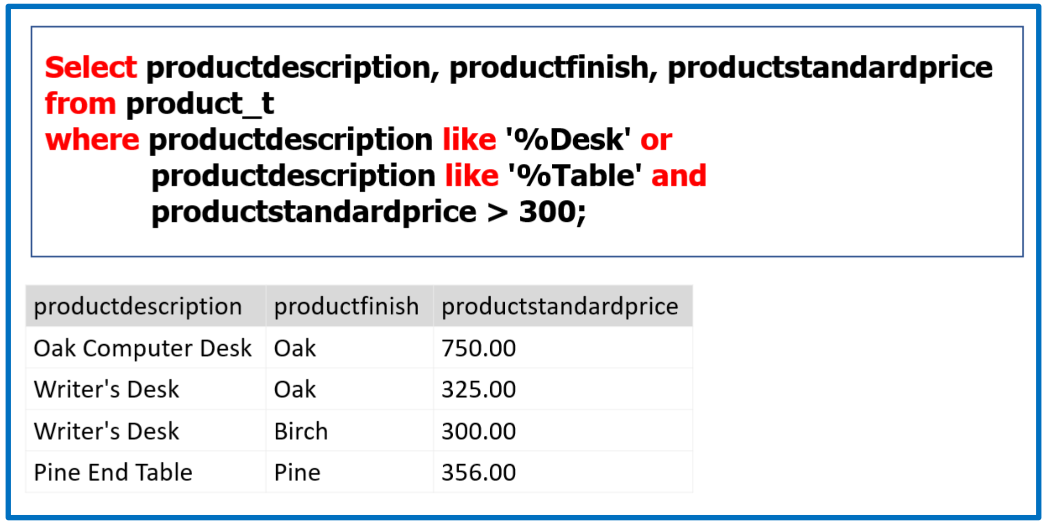
Step 1
Process OR
WHERE ProductDescription LIKE '%Desk' OR ProductDescription LIKE '%Table'
OR
All Desks
All Tables
AND
AND
Products with StandardPrice > $300
Step 2
Process AND
WHERE Result of first process AND StandardPrice >$300

SELECT Example – Boolean Operators

```
Select productdescription, productfinish, productstandardprice
from product_t
where  ( productdescription like '%Desk' OR
          productdescription like '%Table' )  AND
          productstandardprice > 300;
```

productdescription	productfinish	productstandardprice
Oak Computer Desk	Oak	750.00
Writer's Desk	Oak	325.00
Pine End Table	Pine	356.00

Compare the results of two different select statements



```
Select productdescription, productfinish, productstandardprice
from product_t
where productdescription like '%Desk' or
          productdescription like '%Table' and
          productstandardprice > 300;
```

productdescription	productfinish	productstandardprice
Oak Computer Desk	Oak	750.00
Writer's Desk	Oak	325.00
Writer's Desk	Birch	300.00
Pine End Table	Pine	356.00

3.2.4.4. “Is not null” Keyword in Where Clause

```
Select * from product_T;
```

```
 productid | productlineid |    productdescription     | productfinish | productstandardprice | productonhand
-----------+---------------+---------------------------+---------------+----------------------+---------------
         1 |             1 | Cherry End Table          | Cherry        |               175.00 |             0
         2 |             1 | Birch Coffee Tables       | Birch         |               200.00 |             0
         3 |             1 | Oak Computer Desk         | Oak           |               750.00 |             0
         4 |             1 | Entertainment Center      | Cherry        |              1650.00 |             0
         5 |             2 | Writer's Desk             | Oak           |               325.00 |             0
         6 |             1 | 8-Drawer Dresser          | Birch         |               750.00 |             0
         7 |             3 | 48 Bookcase               | Walnut        |               150.00 |             0
         8 |             3 | 48 Bookcase               | Oak           |               175.00 |             0
         9 |             3 | 96 Bookcase               | Walnut        |               225.00 |             0
        10 |             3 | 96 Bookcase               | Oak           |               200.00 |             0
        11 |             1 | 4-Drawer Dresser          | Oak           |               500.00 |             0
        12 |             1 | 8-Drawer Dresser          | Oak           |               800.00 |             0
        13 |             1 | Nightstand                | Cherry        |               150.00 |             0
        14 |             2 | Writer's Desk             | Birch         |               300.00 |             0
        17 |             3 | High Back Leather Chair   | Leather       |               362.00 |             0
        18 |             4 | 6' Grandfather Clock      | Oak           |               890.00 |             0
        19 |             4 | 7' Grandfather Clock      | Oak           |              1100.00 |             0
        20 |             2 | Amoire                    | Walnut        |              1200.00 |             0
        21 |             1 | Pine End Table            | Pine          |               256.00 |             0
        24 |             5 |                           |               |                 0.00 |             0
        25 |             2 |                           |               |                 0.00 |             0
(21 rows)
```

3.2.4.4. “Is not null” Keyword in Where Clause

```
Select * from product_t
where productdescription is not null;
```

you **CANNOT** write either
= NULL
or
<> NULL

```
 productid | productlineid |    productdescription     | productfinish | productstandardprice | productonhand
-----------+---------------+---------------------------+---------------+----------------------+---------------
         1 |             1 | Cherry End Table          | Cherry        |               175.00 |             0
         2 |             1 | Birch Coffee Tables       | Birch         |               200.00 |             0
         3 |             1 | Oak Computer Desk         | Oak           |               750.00 |             0
         4 |             1 | Entertainment Center      | Cherry        |              1650.00 |             0
         5 |             2 | Writer's Desk             | Oak           |               325.00 |             0
         6 |             1 | 8-Drawer Dresser          | Birch         |               750.00 |             0
         7 |             3 | 48 Bookcase               | Walnut        |               150.00 |             0
         8 |             3 | 48 Bookcase               | Oak           |               175.00 |             0
         9 |             3 | 96 Bookcase               | Walnut        |               225.00 |             0
        10 |             3 | 96 Bookcase               | Oak           |               200.00 |             0
        11 |             1 | 4-Drawer Dresser          | Oak           |               500.00 |             0
        12 |             1 | 8-Drawer Dresser          | Oak           |               800.00 |             0
        13 |             1 | Nightstand                | Cherry        |               150.00 |             0
        14 |             2 | Writer's Desk             | Birch         |               300.00 |             0
        17 |             3 | High Back Leather Chair   | Leather       |               362.00 |             0
        18 |             4 | 6' Grandfather Clock      | Oak           |               890.00 |             0
        19 |             4 | 7' Grandfather Clock      | Oak           |              1100.00 |             0
        20 |             2 | Amoire                    | Walnut        |              1200.00 |             0
        21 |             1 | Pine End Table            | Pine          |               256.00 |             0
(19 rows)
```

3.2.4.5. In Operator in Where Clause

Question: Determine information about name, city and state of customers who lives in FL, TX, CA or HI states.

```
Select customername, customercity, customerstate
from customer_t
where customerstate like 'FL' OR customerstate like ‘TX' OR
          customerstate like ‘CA' OR customerstate like ‘HI‘;
```

```
Select customername, customercity, customerstate
from customer_t
where customerstate in ('FL','TX','CA','HI’);
```

Note: The **IN** operator in this example allows you to include rows whose CustomerState value is either FL, TX, CA, or HI.

- Order By Clause
- Aggregation Functions
- Group by Clause

Group By
f(x)
f(x)
f(x)

3.3. SELECT Statement: Order By Clause

Clauses of the SELECT statement:

- SELECT
  - List the columns (and expressions) to be returned from the query
- FROM
  - Indicate the table(s) or view(s) from which data will be obtained
- WHERE
  - Indicate the conditions under which a row will be included in the result
- GROUP BY
  - Indicate categorization of results
- HAVING
  - Indicate the conditions under which a category (group) will be included
- **ORDER BY**
  - **Sorts the result according to specified criteria**

Sorting Results with ORDER BY Clause

Question: Write a query to show customer name, customer city, customer state. Sort the results first by STATE, and within a state by the CUSTOMER NAME.

```
Select customername, customercity, customerstate
from customer_t
order by customerstate, customername;
```

customername	customercity	customerstate
Impressions	Sacramento	CA
Furniture Gallery	Boulder	CO
Contemporary Casuals	Gainesville	FL
Flanigan Furniture	Ft Walton Beach	FL
Wild Bills	Oak Brook	Il
Eastern Furniture	Carteret	NJ
Ikards	Las Cruces	NM
New Furniture	Farmington	NM
A Carpet	Rome	NY
ABC Furniture Co.	Rome	NY
Dunkins Furniture	Syracuse	NY
Home Furnishings	Albany	NY
Value Furnitures	Plano	TX
Janet's Collection	Virginia Beach	VA

Sorting Results with ORDER BY Clause in Ascending and Descending order

Explore: Run these three queries and find the differences in their results table.

What is the default sorting order? Ascending or Descending?

```
Select customername, customercity, customerstate
from customer_t
order by customerstate asc, customername desc;
```

```
Select customername, customercity, customerstate
from customer_t
order by customerstate, customername desc;
```

```
Select customername, customercity, customerstate
from customer_t
order by customerstate desc, customername asc;
```

3.4. SELECT Statement: Group By Clause

Clauses of the SELECT statement:

- SELECT
  - List the columns (and expressions) to be returned from the query
- FROM
  - Indicate the table(s) or view(s) from which data will be obtained
- WHERE
  - Indicate the conditions under which a row will be included in the result
- GROUP BY
  - Indicate categorization of results
- HAVING
  - Indicate the conditions under which a category (group) will be included
- ORDER BY
  - Sorts the result according to specified criteria

3.4.1. Group By

- **Using group by:**
  - ✓ **you can categorize your results into several groups,**
  - ✓ **then analyse the data in each group based on your required information.**

- **Aggregate functions such as *AVG*, *SUM*, *MIN*, *MAX* and *COUNT* can be used to analyse the data in each group**

Note: Before practicing Group By clause, we need to know about Aggregate Function

3.4.1. Aggregate Function: AVG, SUM, MIN, Max and count

Select * from product_T;

```
 productid | productlineid |    productdescription          | productfinish | productstandardprice | productonhand
-----------+---------------+--------------------------------+---------------+----------------------+---------------
         1 |             1 | Cherry End Table               | Cherry        |               175.00 |             0
         2 |             1 | Birch Coffee Tables            | Birch         |               200.00 |             0
         3 |             1 | Oak Computer Desk              | Oak           |               750.00 |             0
         4 |             1 | Entertainment Center           | Cherry        |              1650.00 |             0
         5 |             2 | Writer's Desk                  | Oak           |               325.00 |             0
         6 |             1 | 8-Drawer Dresser               | Birch         |               750.00 |             0
         7 |             3 | 48 Bookcase                    | Walnut        |               150.00 |             0
         8 |             3 | 48 Bookcase                    | Oak           |               175.00 |             0
         9 |             3 | 96 Bookcase                    | Walnut        |               225.00 |             0
        10 |             3 | 96 Bookcase                    | Oak           |               200.00 |             0
        11 |             1 | 4-Drawer Dresser               | Oak           |               500.00 |             0
        12 |             1 | 8-Drawer Dresser               | Oak           |               800.00 |             0
        13 |             1 | Nightstand                     | Cherry        |               150.00 |             0
        14 |             2 | Writer's Desk                  | Birch         |               300.00 |             0
        17 |             3 | High Back Leather Chair        | Leather       |               362.00 |             0
        18 |             4 | 6' Grandfather Clock           | Oak           |               890.00 |             0
        19 |             4 | 7' Grandfather Clock           | Oak           |              1100.00 |             0
        20 |             2 | Amoire                         | Walnut        |              1200.00 |             0
        21 |             1 | Pine End Table                 | Pine          |               256.00 |             0
        24 |             5 |                                |               |                 0.00 |             0
        25 |             2 |                                |               |                 0.00 |             0
(21 rows)
```

3.4.1. Aggregate Function: AVG, SUM, MIN, Max and count

Question: Calculate the average price for all products.

Note: AVG function, short for “average”

```
select avg(productstandardprice) from product_t;

avg
----------------------
 483.7142857142857143
```

```
select round(avg(productstandardprice), 2) from
product_t;

round
----------------------
 483.71
```

3.4.1. Aggregate Function: AVG, SUM, MIN, Max and COUNT

Question: how many products we have?

Note: Using **COUNT** function, we can count the **number of rows** in a **table/group/result table of a select statement.**

```
Select count(*) from product_T;

 count
-------
    21
(1 row)
```

'*' means "the whole row".

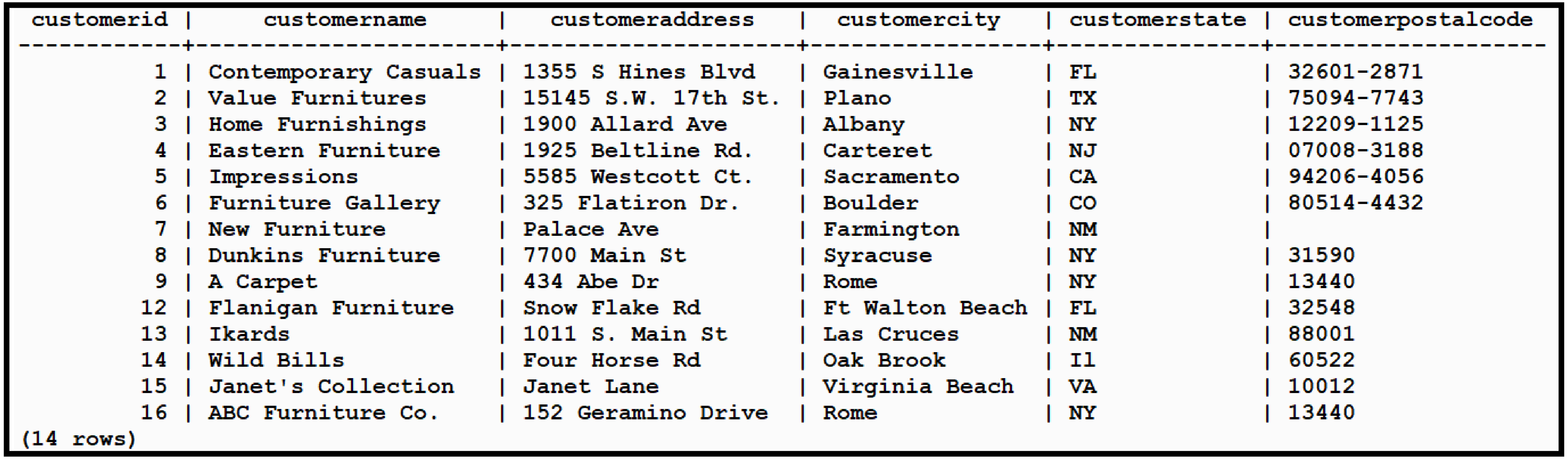
```
Select count(productdescription) from product_T;

 count
-------
    19
(1 row)
```

Null values in "productdescription" are not counted.

3.4.1. Group by

Question: Determine The Number Of Customers In Each State.

Select * from customer_T;

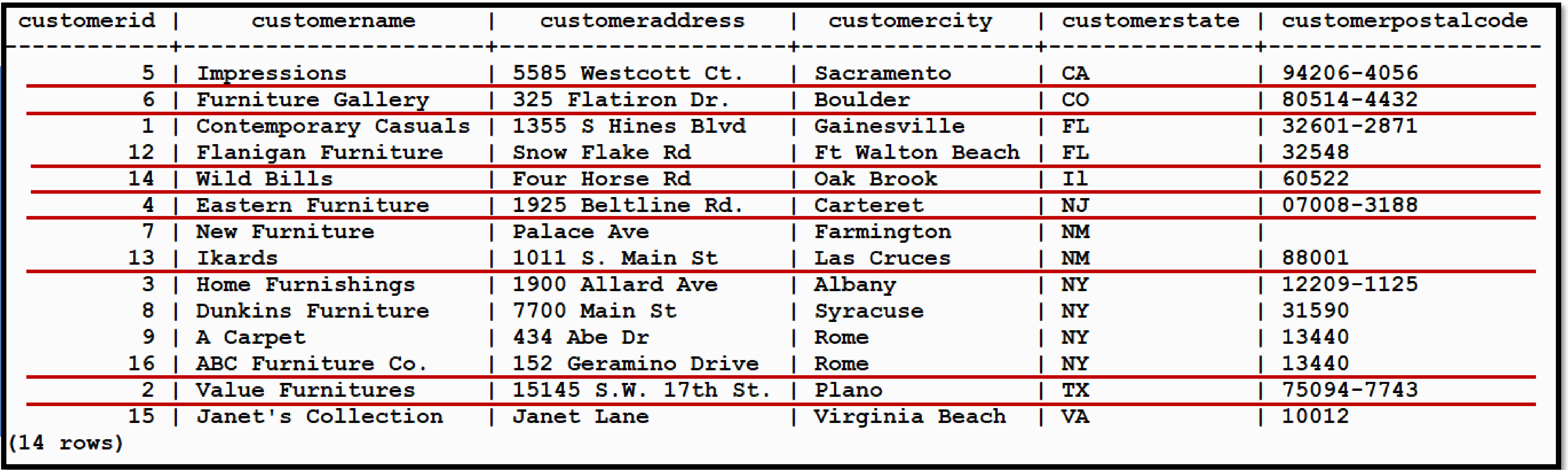


```
 customerid |     customername      |    customeraddress     |   customercity    | customerstate | customerpostalcode
------------+-----------------------+------------------------+-------------------+---------------+--------------------
          1 | Contemporary Casuals  | 1355 S Hines Blvd      | Gainesville       | FL            | 32601-2871
          2 | Value Furnitures      | 15145 S.W. 17th St.    | Plano             | TX            | 75094-7743
          3 | Home Furnishings      | 1900 Allard Ave        | Albany            | NY            | 12209-1125
          4 | Eastern Furniture     | 1925 Beltline Rd.      | Carteret          | NJ            | 07008-3188
          5 | Impressions           | 5585 Westcott Ct.      | Sacramento        | CA            | 94206-4056
          6 | Furniture Gallery     | 325 Flatiron Dr.       | Boulder           | CO            | 80514-4432
          7 | New Furniture         | Palace Ave             | Farmington        | NM            |
          8 | Dunkins Furniture     | 7700 Main St           | Syracuse          | NY            | 31590
          9 | A Carpet              | 434 Abe Dr             | Rome              | NY            | 13440
         12 | Flanigan Furniture    | Snow Flake Rd          | Ft Walton Beach   | FL            | 32548
         13 | Ikards                | 1011 S. Main St        | Las Cruces        | NM            | 88001
         14 | Wild Bills            | Four Horse Rd          | Oak Brook         | Il            | 60522
         15 | Janet's Collection    | Janet Lane             | Virginia Beach    | VA            | 10012
         16 | ABC Furniture Co.     | 152 Geramino Drive     | Rome              | NY            | 13440
(14 rows)
```

3.4.2. Group by Example: Determine The Number Of Customers In Each State.

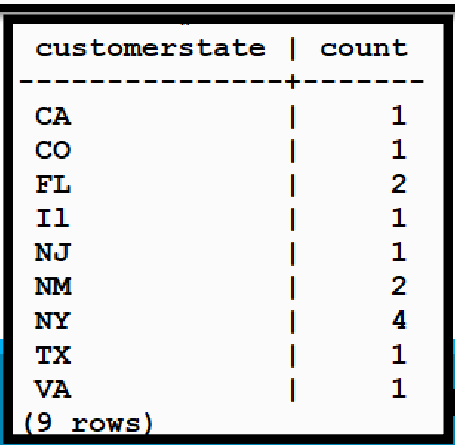
```
Select customerstate, count(customerstate) from customer_t group by customerstate

Order by customerstate;
```

Intermediate results tables



```
 customerid |      customername      |   customeraddress     |  customercity     | customerstate | customerpostalcode
------------+------------------------+-----------------------+-------------------+---------------+--------------------
          5 | Impressions            | 5585 Westcott Ct.     | Sacramento        | CA            | 94206-4056
          6 | Furniture Gallery      | 325 Flatiron Dr.      | Boulder           | CO            | 80514-4432
          1 | Contemporary Casuals   | 1355 S Hines Blvd     | Gainesville       | FL            | 32601-2871
         12 | Flanigan Furniture     | Snow Flake Rd         | Ft Walton Beach   | FL            | 32548
         14 | Wild Bills             | Four Horse Rd         | Oak Brook         | Il            | 60522
          4 | Eastern Furniture      | 1925 Beltline Rd.     | Carteret          | NJ            | 07008-3188
          7 | New Furniture          | Palace Ave            | Farmington        | NM            |
         13 | Ikards                 | 1011 S. Main St       | Las Cruces        | NM            | 88001
          3 | Home Furnishings       | 1900 Allard Ave       | Albany            | NY            | 12209-1125
          8 | Dunkins Furniture      | 7700 Main St          | Syracuse          | NY            | 31590
          9 | A Carpet               | 434 Abe Dr            | Rome              | NY            | 13440
         16 | ABC Furniture Co.      | 152 Geramino Drive    | Rome              | NY            | 13440
          2 | Value Furnitures       | 15145 S.W. 17th St.   | Plano             | TX            | 75094-7743
         15 | Janet's Collection     | Janet Lane            | Virginia Beach    | VA            | 10012
(14 rows)
```

Results tables



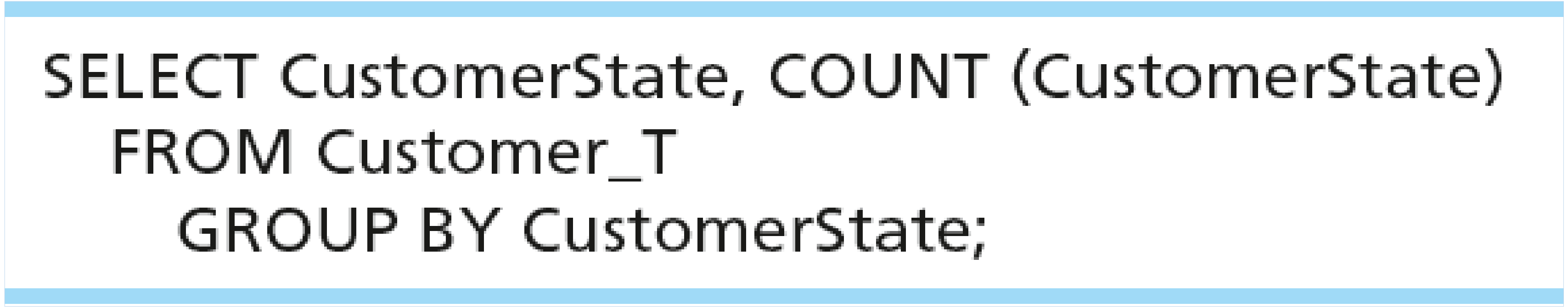
```
 customerstate | count
---------------+-------
 CA            |     1
 CO            |     1
 FL            |     2
 Il            |     1
 NJ            |     1
 NM            |     2
 NY            |     4
 TX            |     1
 VA            |     1
(9 rows)
```

3.4.2.Categorizing Results Using GROUP BY Clause

- **Scalar aggregate**: single value returned from SQL query with aggregate function

```
select avg(productstandardprice) from product_t;
```

- **Vector aggregate**: multiple values returned from SQL query with aggregate function (via GROUP BY)



```
SELECT CustomerState, COUNT (CustomerState)
  FROM Customer_T
    GROUP BY CustomerState;
```

Note: You can use single-value fields with aggregate functions if they are included in the GROUP BY clause

3.4.3.Group by clause: Rule 1

Note: You can use single-value fields with aggregate functions if they are included in the GROUP BY clause

```
Select customerstate, count(customerstate)
from customer_t
group by customerstate;
```

```
 customerstate | count
---------------+-------
 NY              |     4
 CO              |      1
 TX              |     1
 CA              |     1
 FL              |     2
 NM              |      2
 VA              |     1
 NJ              |      1
 Il              |      1
(9 rows)
```

Rule 1: The first one (or more) columns nominated in the "select" **must also** be nominated in the "GROUP BY".

3.4.4. Group by clause: Rule 2

Note: You can use single-value fields with aggregate functions if they are included in the GROUP BY clause

```
Select customerstate, count(customerstate)
from customer_t
group by customerstate;
```

Rule 2: The remaining columns nominated in the "select" **must be** aggregate functions (often "count").

```
 customerstate | count
---------------+-------
 NY            |     4
 CO            |     1
 TX            |     1
 CA            |     1
 FL            |     2
 NM            |     2
 VA            |     1
 NJ            |     1
 Il            |     1
(9 rows)
```

3.5. SELECT Statement: Having

Clauses of the SELECT statement:

- SELECT
  - List the columns (and expressions) to be returned from the query
- FROM
  - Indicate the table(s) or view(s) from which data will be obtained
- WHERE
  - Indicate the conditions under which a row will be included in the result
- GROUP BY
  - Indicate categorization of results
- HAVING
  - Indicate the conditions under which a category (group) will be included
- ORDER BY
  - Sorts the result according to specified criteria

3.5. Qualifying Categories in the results table Using the HAVING Clause

Question: Select the states with more than one customer.

```
SELECT CustomerState, COUNT (CustomerState)
  FROM Customer_T
    GROUP BY CustomerState
    HAVING COUNT (CustomerState) > 1;
```

- **HAVING** after group by works like a **WHERE** clause, but HAVING operates on groups (categories), not on individual rows.
- Here, only those groups with total numbers greater than 1 will be included in final result.

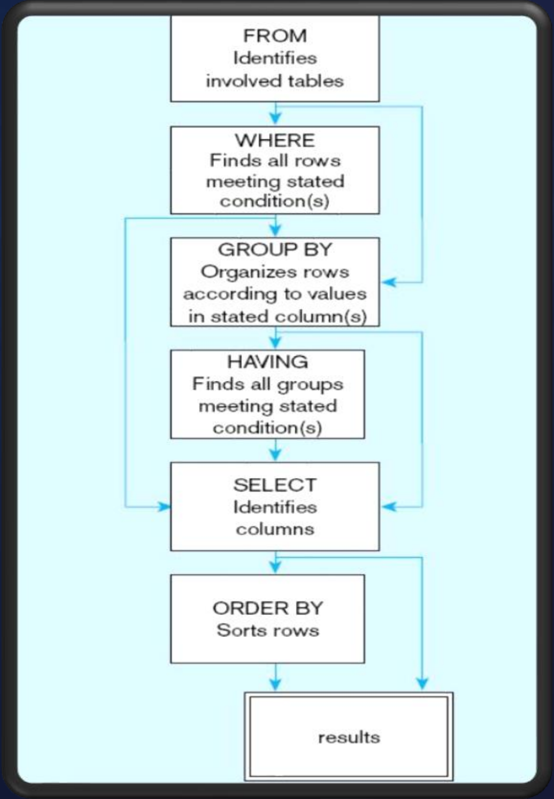
3.5. Having Clause

Question: Select the states with more than one customer.

```
Select customerstate, count(customerstate)
from customer_t
group by customerstate
having count(customerstate) > 1;
```

```
 customerstate | count
---------------+-------
 NY            |     4
 FL            |     2
 NM            |     2
(3 rows)
```

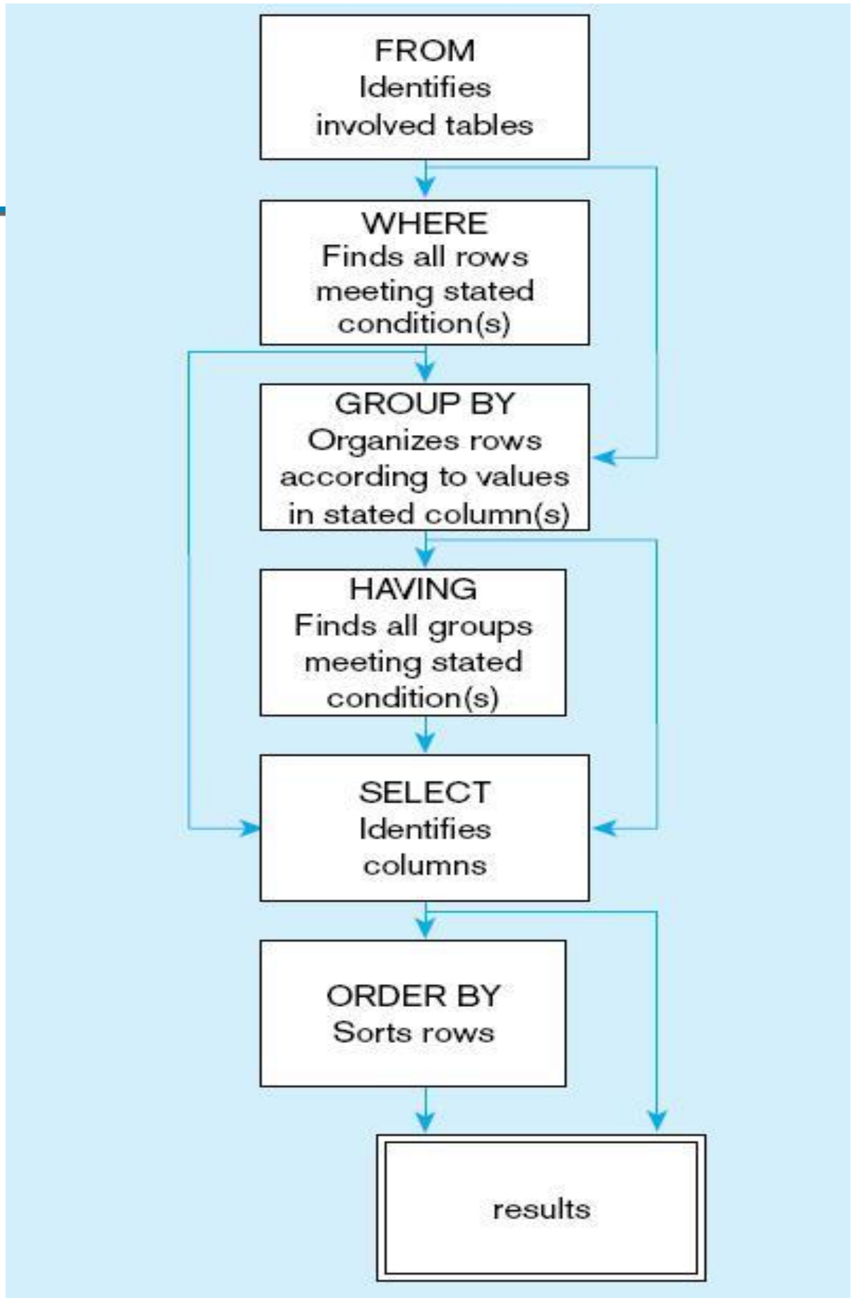
SQL statement processing order



FROM
Identifies involved tables
WHERE
Finds all rows meeting stated condition(s)
GROUP BY
Organizes rows according to values in stated column(s)
HAVING
Finds all groups meeting stated condition(s)
SELECT
Identifies columns
ORDER BY
Sorts rows
results

3.6. SQL statement processing order (Figure 6-10)

```
Select column1, column2
From Table1
Where [Condition on rows]
Group by column1, column2
Having [Condition on groups]
Order by column1
```


FROM
Identifies involved tables
WHERE
Finds all rows meeting stated condition(s)
GROUP BY
Organizes rows according to values in stated column(s)
HAVING
Finds all groups meeting stated condition(s)
SELECT
Identifies columns
ORDER BY
Sorts rows
results

SQL statement processing order

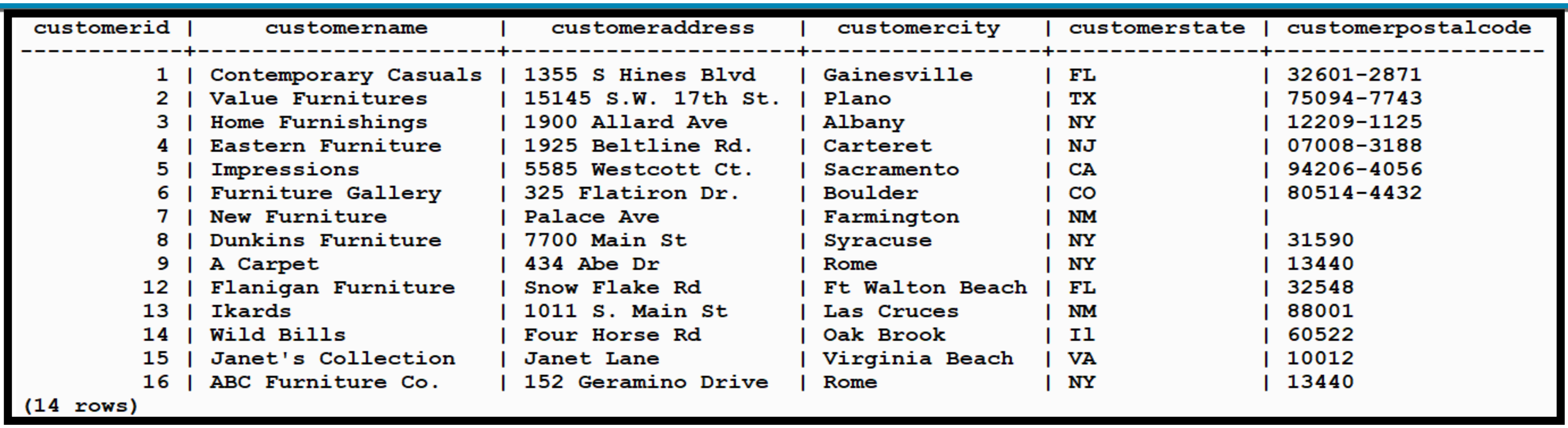
- As each clause is processed, an intermediate results table is produced that will be used for the next clause.
- Users do not see the intermediate results tables; they only see the final results table.

Example:

- **Provide a list of customer state and the number of customers in each state with postal code between 30000 and 80000. Just list the states with less than two customers. Sort the results based on customer state.**

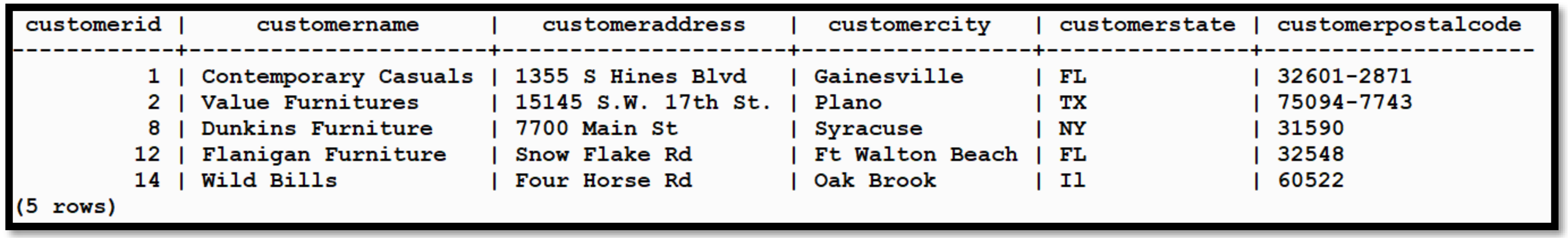
```
select customerstate, count(customerstate) from customer_t
where customerpostalcode between '30000' and '80000'
group by customerstate having count(*)<2 order by customerstate;
```

From



```
 customerid |     customername      |    customeraddress    |   customercity    | customerstate | customerpostalcode
------------+-----------------------+-----------------------+-------------------+---------------+--------------------
          1 | Contemporary Casuals | 1355 S Hines Blvd     | Gainesville       | FL            | 32601-2871
          2 | Value Furnitures      | 15145 S.W. 17th St.   | Plano             | TX            | 75094-7743
          3 | Home Furnishings      | 1900 Allard Ave       | Albany            | NY            | 12209-1125
          4 | Eastern Furniture     | 1925 Beltline Rd.     | Carteret          | NJ            | 07008-3188
          5 | Impressions           | 5585 Westcott Ct.     | Sacramento        | CA            | 94206-4056
          6 | Furniture Gallery     | 325 Flatiron Dr.      | Boulder           | CO            | 80514-4432
          7 | New Furniture         | Palace Ave            | Farmington        | NM            |
          8 | Dunkins Furniture     | 7700 Main St          | Syracuse          | NY            | 31590
          9 | A Carpet              | 434 Abe Dr            | Rome              | NY            | 13440
         12 | Flanigan Furniture    | Snow Flake Rd         | Ft Walton Beach   | FL            | 32548
         13 | Ikards                | 1011 S. Main St       | Las Cruces        | NM            | 88001
         14 | Wild Bills            | Four Horse Rd         | Oak Brook         | Il            | 60522
         15 | Janet's Collection    | Janet Lane            | Virginia Beach    | VA            | 10012
         16 | ABC Furniture Co.     | 152 Geramino Drive    | Rome              | NY            | 13440
(14 rows)
```

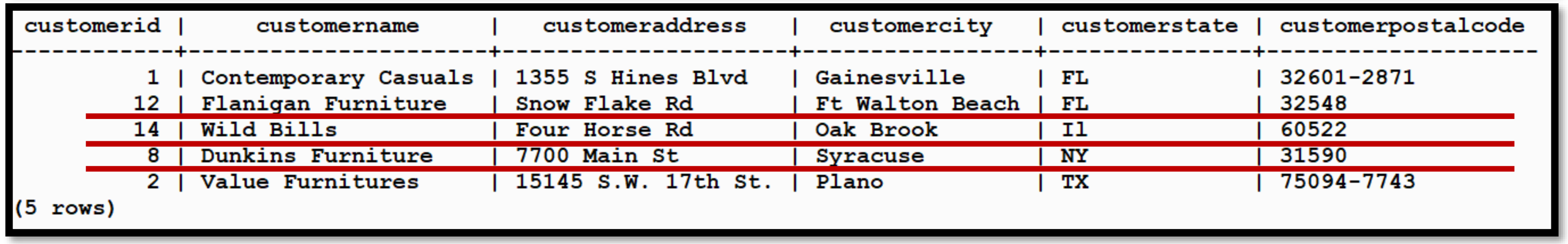
where



```
 customerid |     customername      |    customeraddress    |   customercity    | customerstate | customerpostalcode
------------+-----------------------+-----------------------+-------------------+---------------+--------------------
          1 | Contemporary Casuals | 1355 S Hines Blvd     | Gainesville       | FL            | 32601-2871
          2 | Value Furnitures      | 15145 S.W. 17th St.   | Plano             | TX            | 75094-7743
          8 | Dunkins Furniture     | 7700 Main St          | Syracuse          | NY            | 31590
         12 | Flanigan Furniture    | Snow Flake Rd         | Ft Walton Beach   | FL            | 32548
         14 | Wild Bills            | Four Horse Rd         | Oak Brook         | Il            | 60522
(5 rows)
```

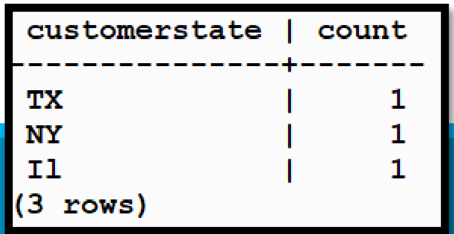
group by

having



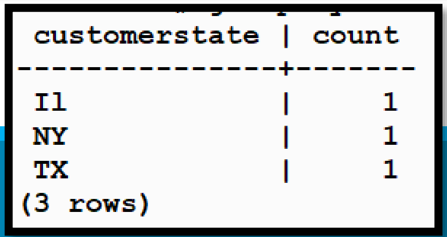
```
 customerid |     customername      |    customeraddress    |   customercity    | customerstate | customerpostalcode
------------+-----------------------+-----------------------+-------------------+---------------+--------------------
          1 | Contemporary Casuals | 1355 S Hines Blvd     | Gainesville       | FL            | 32601-2871
         12 | Flanigan Furniture    | Snow Flake Rd         | Ft Walton Beach   | FL            | 32548
         14 | Wild Bills            | Four Horse Rd         | Oak Brook         | Il            | 60522
          8 | Dunkins Furniture     | 7700 Main St          | Syracuse          | NY            | 31590
          2 | Value Furnitures      | 15145 S.W. 17th St.   | Plano             | TX            | 75094-7743
(5 rows)
```

select



```
 customerstate | count
---------------+-------
 TX            |     1
 NY            |     1
 Il            |     1
(3 rows)
```

order by



```
 customerstate | count
---------------+-------
 Il            |     1
 NY            |     1
 TX            |     1
(3 rows)
```

CREATE VIEW

Note: This section is also provided in Module 1

3.7. Views

- Tables are used to store data physically in database correspond to relations in logical database design.
- Using SQL queries it is possible to create **virtual table** or **dynamic views.**
- **virtual table** or **dynamic views** can be manipulated like tables.

3.7. Create View

- **Create View syntax:**

```
Create View View_Name as Select_Statement;
```

Example:

```
Create view P as select * from product_t;
```

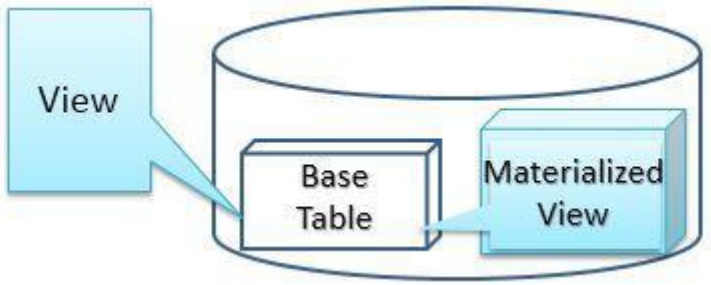
Now P can be used as a table. Run the following query:

```
Select * from P;
```

3.7. Views

- Views provide users-controlled access to tables.
- Base Table containing the raw data.
- Dynamic View
  - A “virtual table” created dynamically upon request by a user.
  - No data actually stored; instead, data from base table made available to user.
  - Results are provided based on a SQL SELECT statement on base tables or other views.
- Materialized View
  - Copy or replication of data.
  - Data actually stored.
  - Must be refreshed periodically to match corresponding base tables.

Key Differences Between Dynamic View and Materialized View



1. The basic difference between View and Materialized View is that Views are **not stored** physically on the disk. On the other hands, Materialized Views are **stored** on the disc.
2. View can be defined as a **virtual table** created as a result of the query expression. However, Materialized View is a **physical copy**, picture or snapshot of the base table.
3. A view is always **updated** as the query creating View executes each time the View is used. On the other hands, Materialized View is updated **manually** or by applying **triggers** to it.
4. Materialized View responds **faster** than View as the Materialized View is precomputed.
5. Materialized View **utilizes** the **memory space** as it stored on the disk whereas, the View is just a **display** hence it do not require memory space.

Reference

3.7. Advantages of Dynamic Views

- Simplify query commands
- Assist with data security (but don't rely on views for security, there are more important security measures)
- Enhance programming productivity
- Contain most current base table data
- Use little storage space
- Provide customized view for user
- Establish physical data independence

3.7. Disadvantages of Dynamic Views

- Use processing time each time view is referenced
- May or may not be directly updateable

Summary (main information)

- Clauses of the SELECT statement:
  - **SELECT** list the columns (and expressions) to be returned from the query
  - **FROM** indicate the table(s) or view(s) from which data will be obtained
  - **WHERE** (Comparison operators, AND, OR, is not null, in/not in, between) indicate the conditions under which a row will be included in the result
  - **GROUP BY** (using aggregate functions AVG, MIN, MAX, SUM and COUNT) indicate categorization of results
  - **HAVING** indicate the conditions under which a category (group) will be included
  - **ORDER BY** Sorts the result according to specified criteria
- Create View