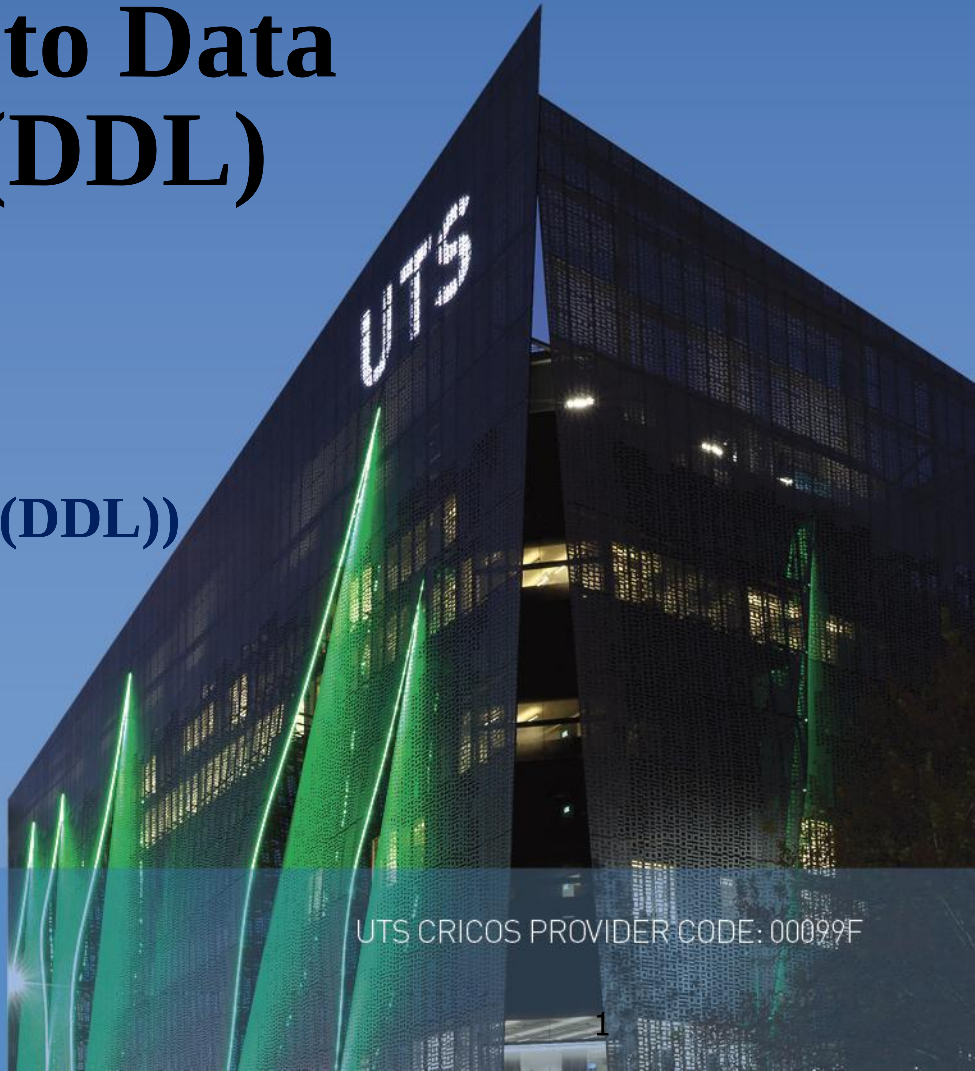
lecture 2: Introduction to Data Definition Language (DDL)

Main reference:

Modern Database Management, 11th Edition
Chapter 6: Introduction to SQL (Data Definition Language (DDL))

Subject Coordinator and Instructor:

Dr. Danna (Fahimeh) Ramezani

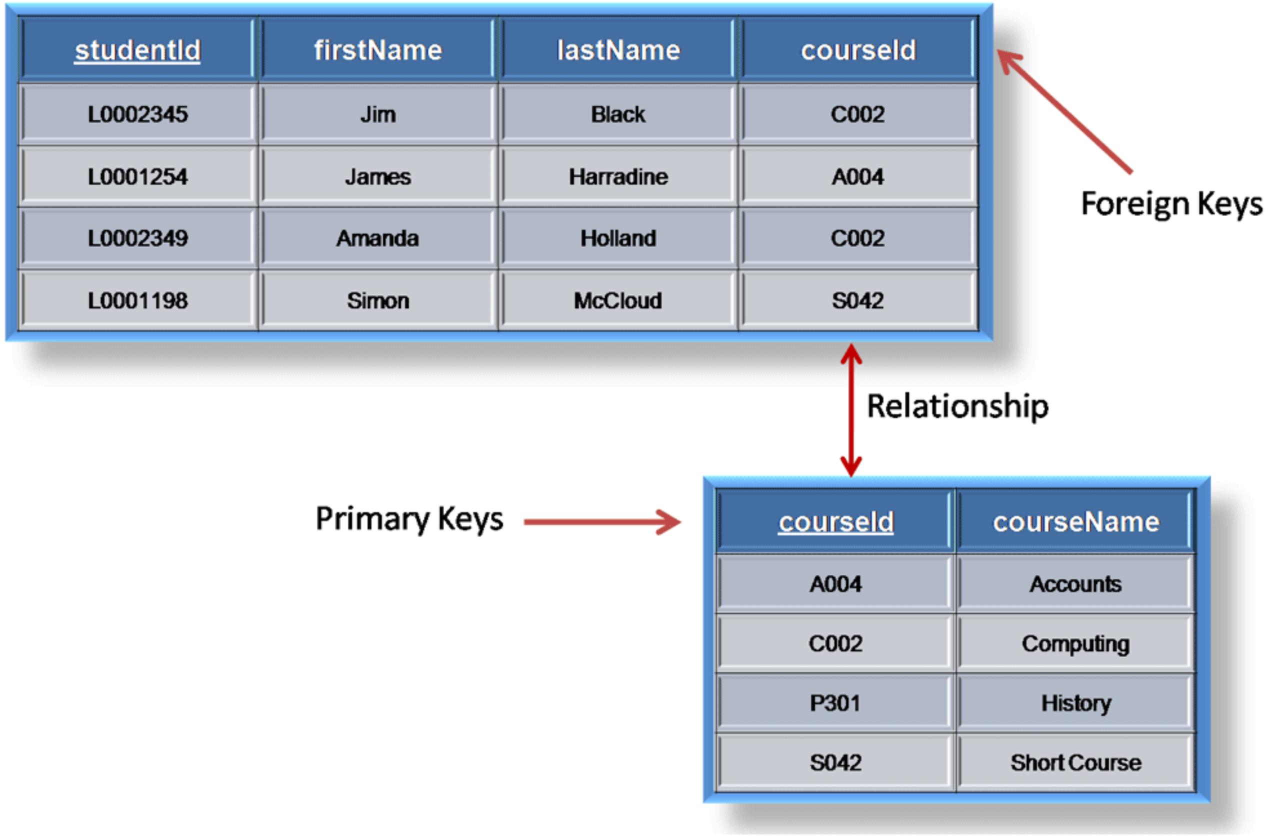


Subject Overview

- **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**

Relational Database Management Systems (RDBMS)

Manages data as a **collection of tables** in which all relationships are represented by **common values in related tables**



studentId	firstName	lastName	courseId
L0002345	Jim	Black	C002
L0001254	James	Harradine	A004
L0002349	Amanda	Holland	C002
L0001198	Simon	McCloud	S042

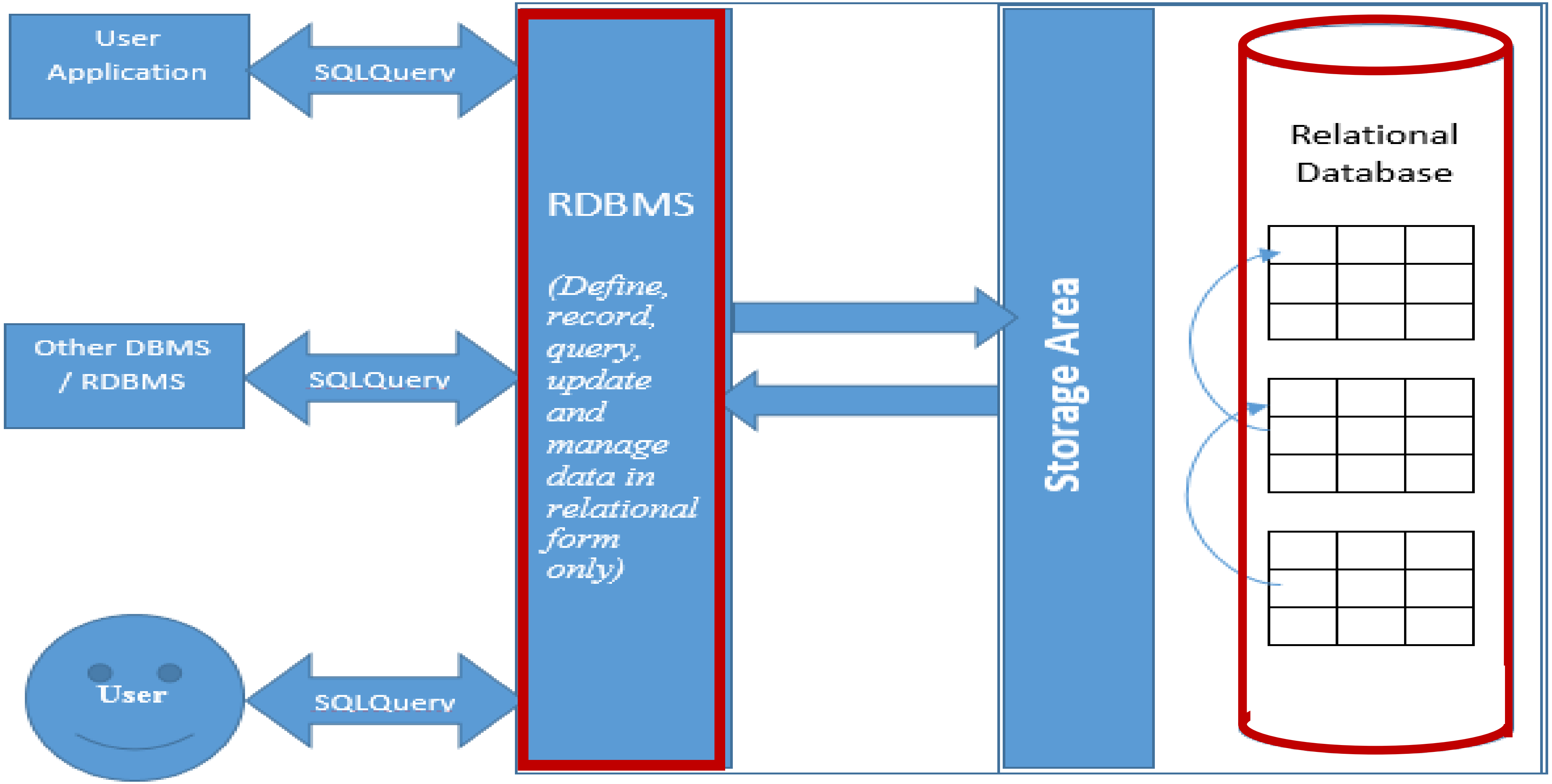
Foreign Keys

Relationship

Primary Keys

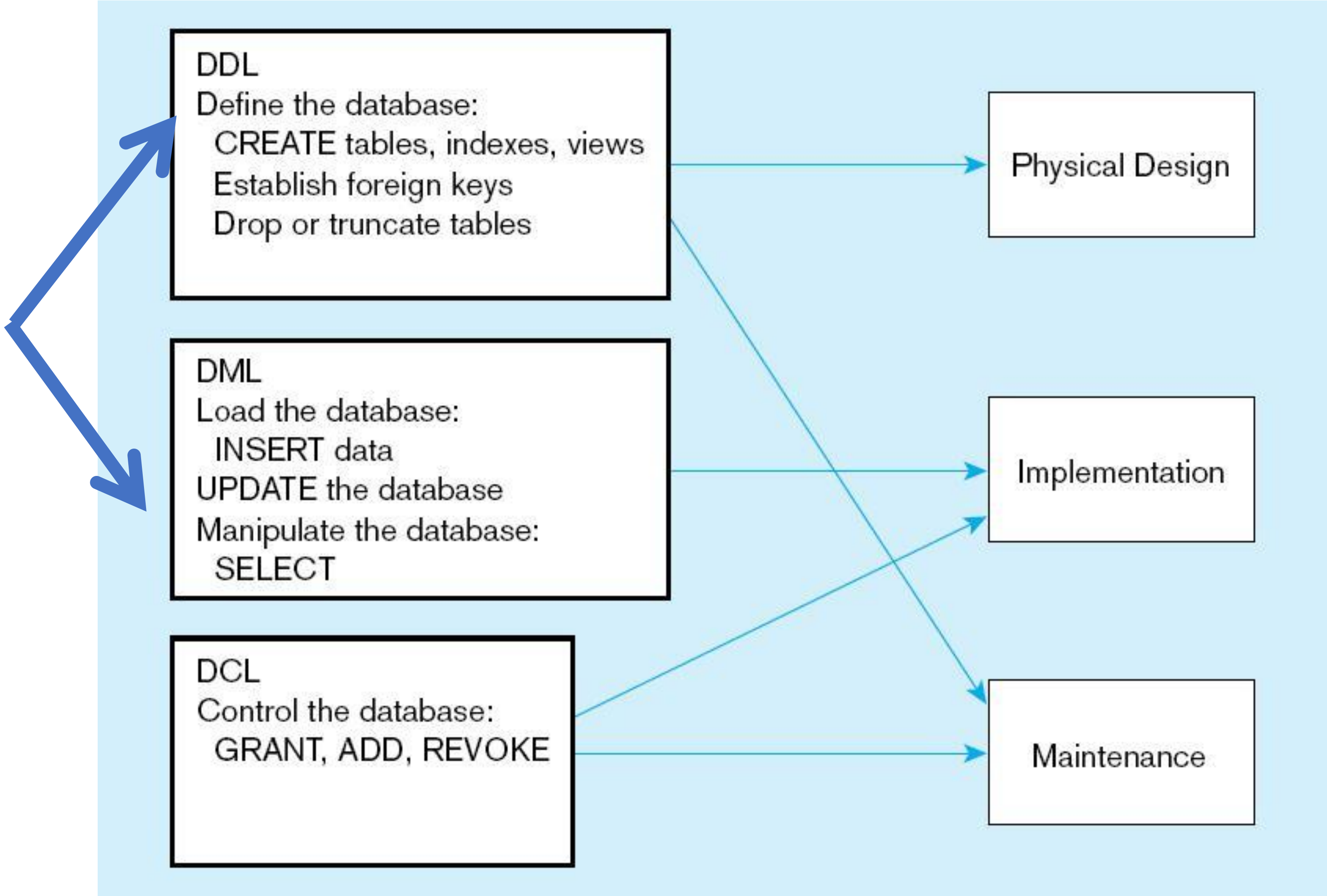
courseId	courseName
A004	Accounts
C002	Computing
P301	History
S042	Short Course

Relational Database Management Systems (RDBMS) (cont.)



SQL Environment: DDL, DML, DCL, and the database development process (Figure 6-4)

focus of SQL lectures



DDL
Define the database:
CREATE tables, indexes, views
Establish foreign keys
Drop or truncate tables

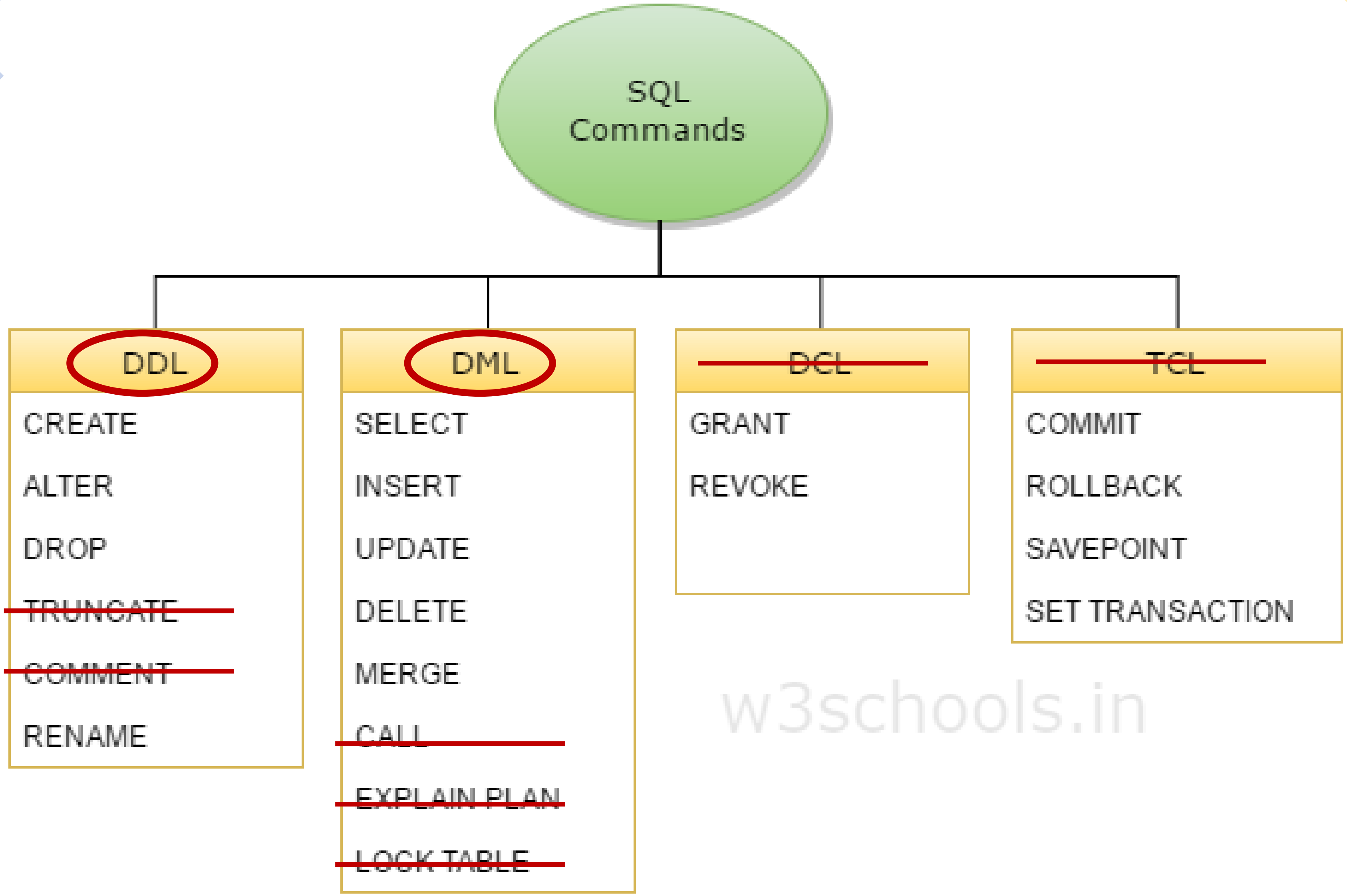
DML
Load the database:
INSERT data
UPDATE the database
Manipulate the database:
SELECT

DCL
Control the database:
GRANT, ADD, REVOKE

Physical Design

Implementation

Maintenance



Note: we will cover some of these statements in this subject.

Image Reference: https://stackoverflow.com/questions/2578194/what-are-ddl-and-dml

Some notes before you start …

- The context of the slides with BLUE tittle are provided for your **self-study** and **will not be part of your assessments.**

- Data Definition Language (DDL) that is related to create, drop, and modify a table, and part of the Data Manipulation Language (DML) such as insert, update and delete statements, **will NOT be part of your SQL online test**.

- You will need this information to create tables and complete the **Assessment Challenge** and **Part D of your assignment.**

- We also use DDL in our lectures to show the how a designed ERD is related to its corresponding database.

SQL Data Types Samples (You will use more data types in your work)

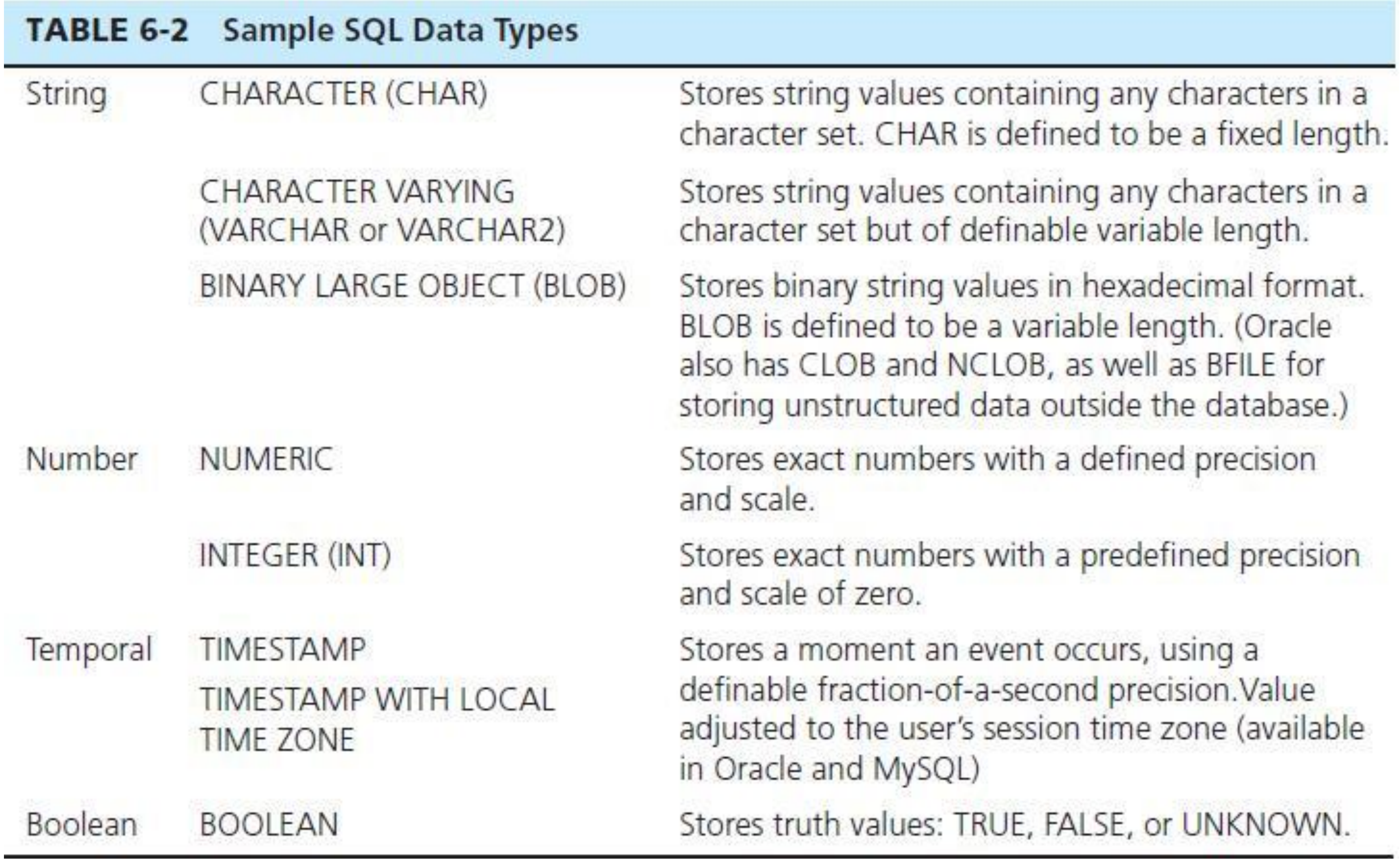


TABLE 6-2 Sample SQL Data Types

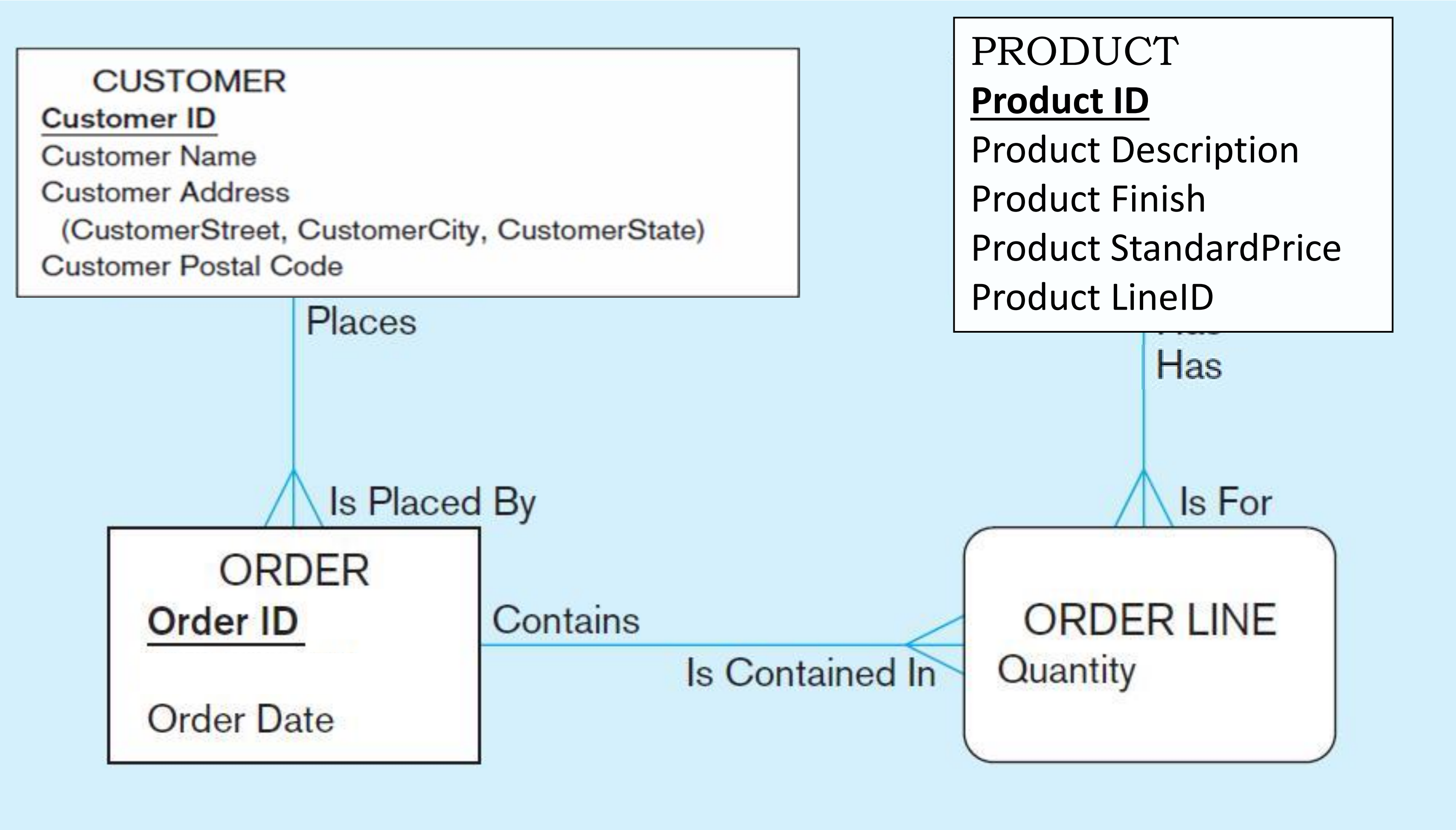
String	CHARACTER (CHAR)	Stores string values containing any characters in a character set. CHAR is defined to be a fixed length.
	CHARACTER VARYING (VARCHAR or VARCHAR2)	Stores string values containing any characters in a character set but of definable variable length.
	BINARY LARGE OBJECT (BLOB)	Stores binary string values in hexadecimal format. BLOB is defined to be a variable length. (Oracle also has CLOB and NCLOB, as well as BFILE for storing unstructured data outside the database.)
Number	NUMERIC	Stores exact numbers with a defined precision and scale.
	INTEGER (INT)	Stores exact numbers with a predefined precision and scale of zero.
Temporal	TIMESTAMP TIMESTAMP WITH LOCAL TIME ZONE	Stores a moment an event occurs, using a definable fraction-of-a-second precision.Value adjusted to the user's session time zone (available in Oracle and MySQL)
Boolean	BOOLEAN	Stores truth values: TRUE, FALSE, or UNKNOWN.

Data Definition Language (DDL):

In the context of SQL, **data definition** or **data description language** (**DDL**) is a syntax for **creating** and **modifying** database objects such as **tables**, and indices.

DDL statements are similar to a computer programming language for defining data structures, especially database schemas.

The following slides create tables for this enterprise data model



CUSTOMER
Customer ID
Customer Name
Customer Address
(CustomerStreet, CustomerCity, CustomerState)
Customer Postal Code

PRODUCT
Product ID
Product Description
Product Finish
Product StandardPrice
Product LineID

ORDER
Order ID
Order Date

ORDER LINE
Quantity

(from Chapter 1, Figure 1-3)

Steps in Table Creation

- Identify data types for attributes
- Identify columns that can and cannot be null
- Identify columns that must be unique (candidate keys)

  This step is not required in this stage. You will know more about candidate key in week 6*.

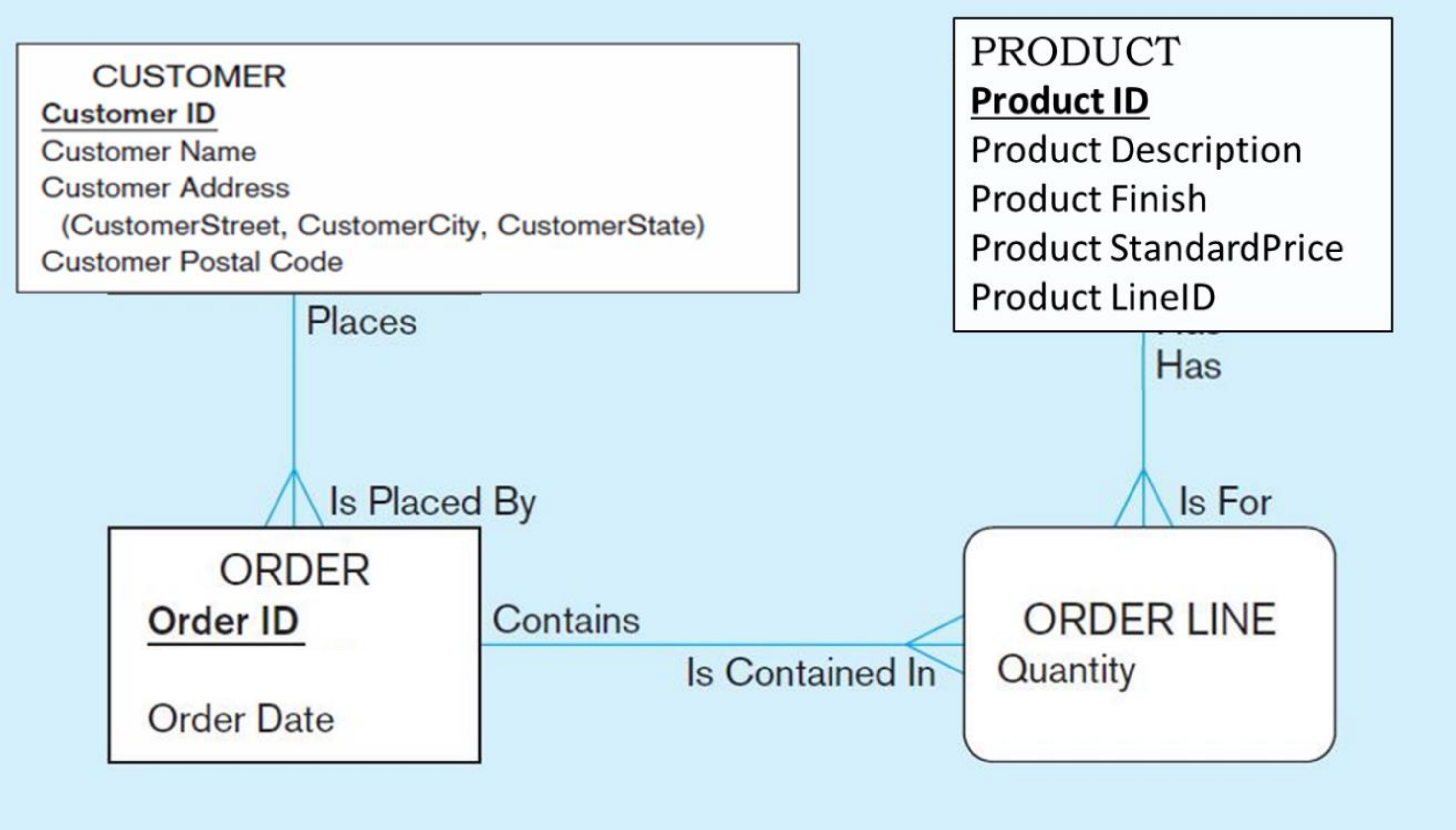
- Identify primary key–foreign key mates
- Determine default values (if it is required)
- Identify constraints on columns (domain specifications)

  This step is not required in this stage. You will know more about domain specifications key in week 5*.

- Create the table and associated indexes

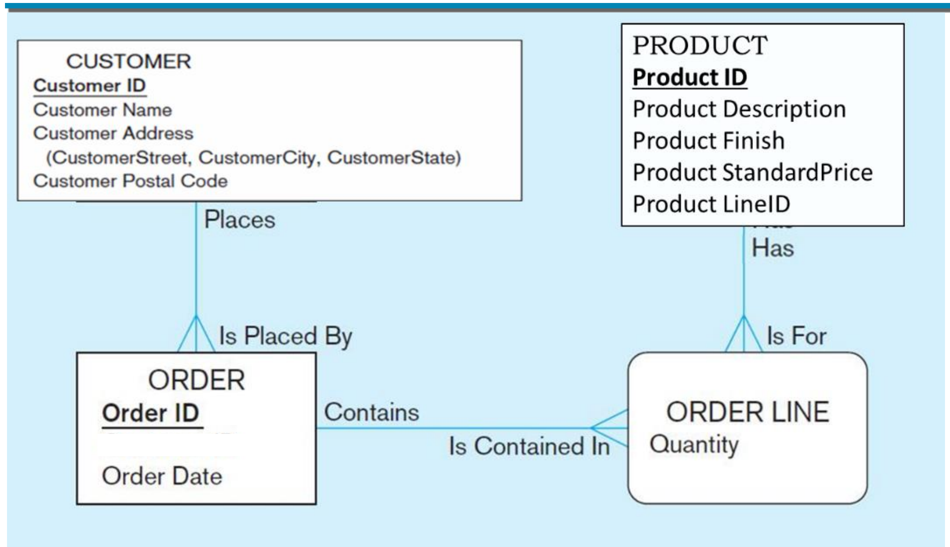
*Note: Considering that we are revising the lecture materials, the other of the concepts related to the specified weeks may change

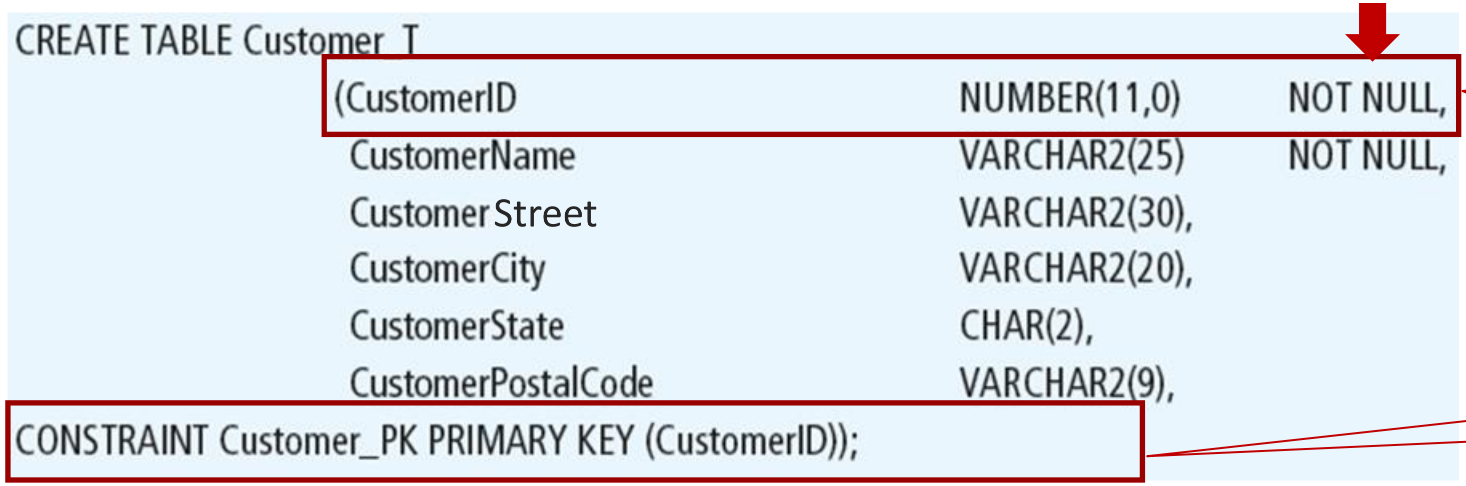
Create Table



Create CUSTOMER Table

1. Identify data types for attributes
2. Identify columns that can and cannot be null
3. Identify columns that must be unique (candidate keys)
4. Identify primary key–foreign key mates
5. Determine default values
6. Identify constraints on columns (domain specifications)
7. Create the table and associated indexes





```
CREATE TABLE Customer_T
          (CustomerID                         NUMBER(11,0)      NOT NULL,
           CustomerName                       VARCHAR2(25)      NOT NULL,
           CustomerStreet                     VARCHAR2(30),
           CustomerCity                       VARCHAR2(20),
           CustomerState                      CHAR(2),
           CustomerPostalCode                 VARCHAR2(9),
CONSTRAINT Customer_PK PRIMARY KEY (CustomerID));
```

Primary keys can never have NULL values

PK Constraint

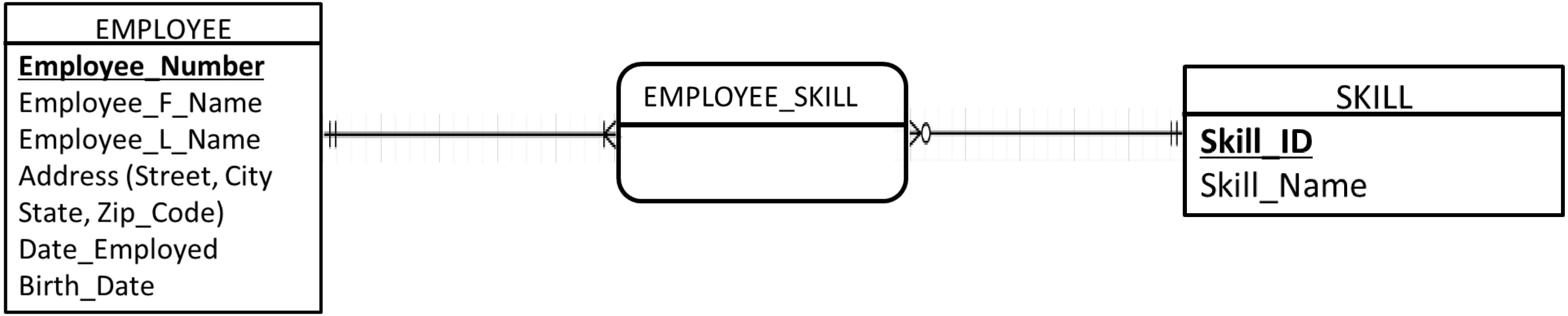
Class Activity 2.1:

Create table SKILL.

BR1: The system needs to provide a unique number for each employee, and collect employees' personal information including name, home address, date of birth, and employment date.

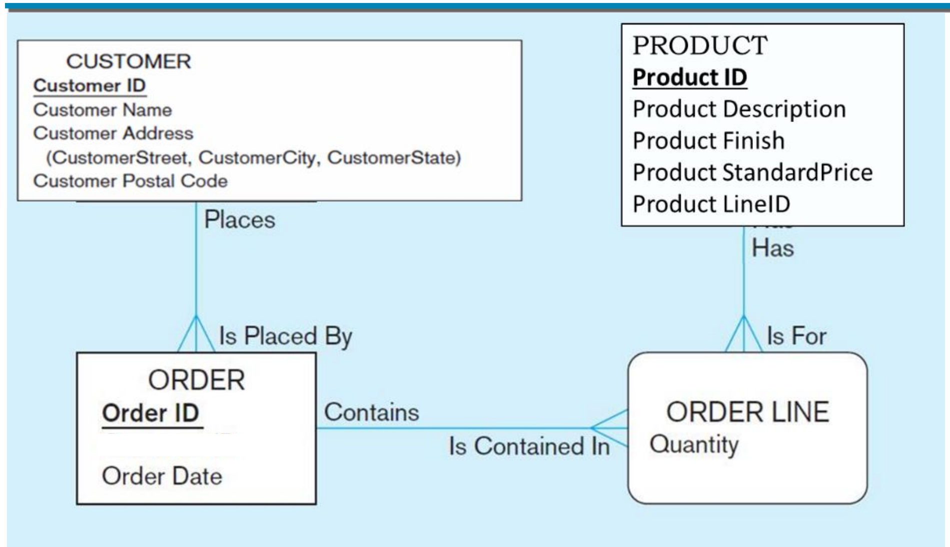
BR2: An employee needs to have experience with one or more general purpose programming languages including: Java, C/C++, C#.

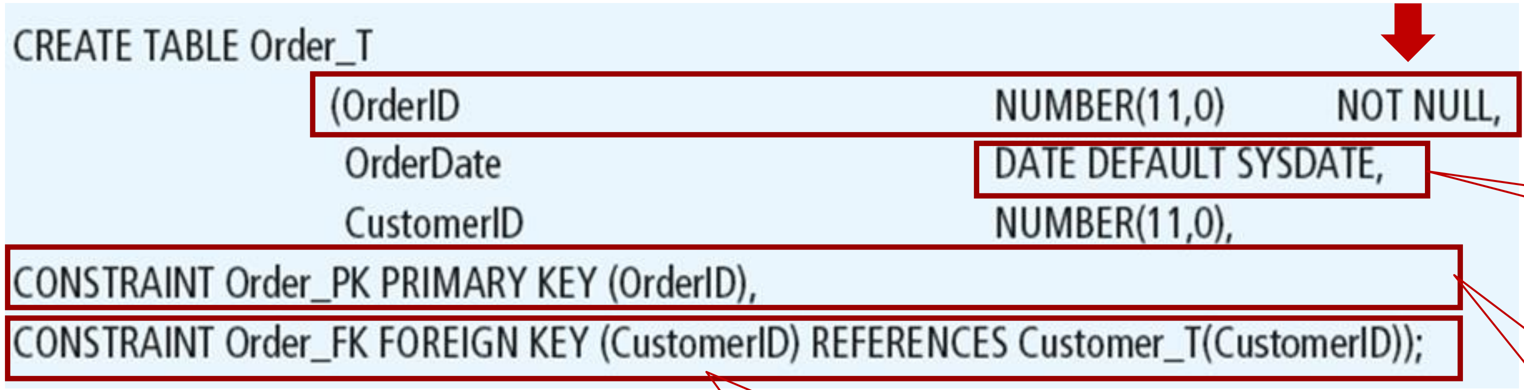
A1: Any skill can be taken by any number of employees.



Create ORDER Table

1. Identify data types for attributes
2. Identify columns that can and cannot be null
3. Identify columns that must be unique (candidate keys)
4. Identify primary key–foreign key mates
5. Determine default values
6. Identify constraints on columns (domain specifications)
7. Create the table and associated indexes





```
CREATE TABLE Order_T
            (OrderID                          NUMBER(11,0)        NOT NULL,
             OrderDate                        DATE DEFAULT SYSDATE,
             CustomerID                       NUMBER(11,0),
CONSTRAINT Order_PK PRIMARY KEY (OrderID),
CONSTRAINT Order_FK FOREIGN KEY (CustomerID) REFERENCES Customer_T(CustomerID));
```

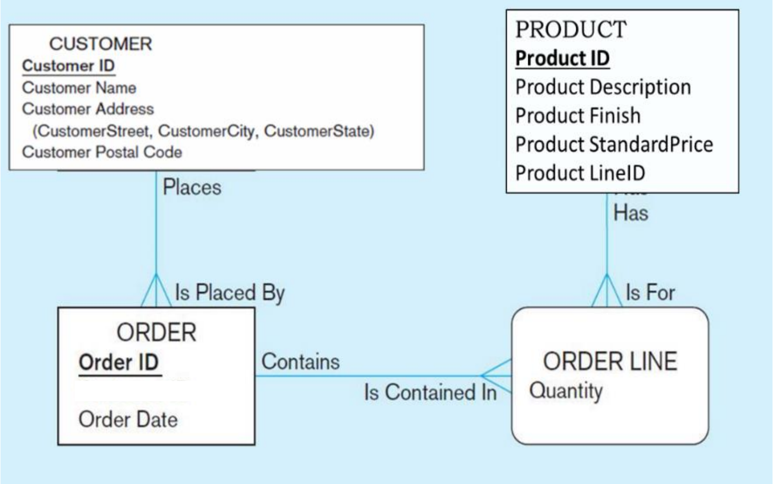
Primary keys can never have NULL values

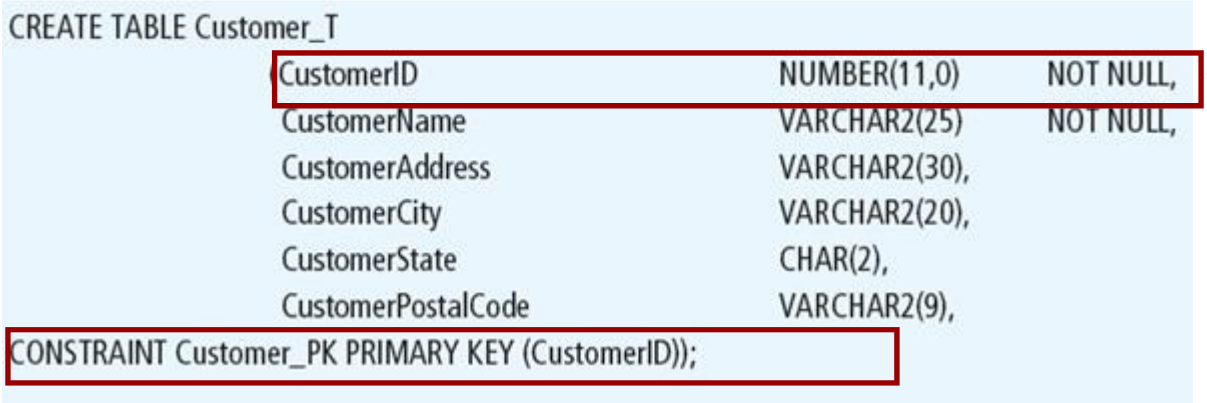
Default Values

Foreign key of dependent table

PK Constraint

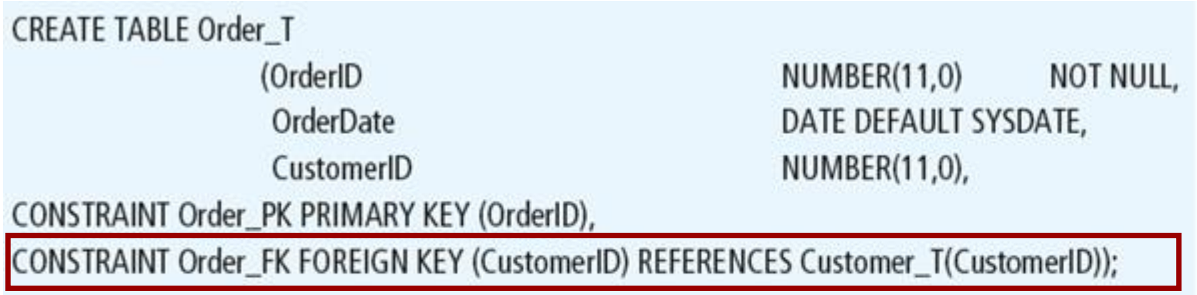
Create ORDER Table … See the relationship with CUSTOMER Table





```
CREATE TABLE Customer_T
              (CustomerID            NUMBER(11,0)     NOT NULL,
               CustomerName          VARCHAR2(25)     NOT NULL,
               CustomerAddress       VARCHAR2(30),
               CustomerCity          VARCHAR2(20),
               CustomerState         CHAR(2),
               CustomerPostalCode    VARCHAR2(9),
CONSTRAINT Customer_PK PRIMARY KEY (CustomerID));
```

Primary key of parent table



```
CREATE TABLE Order_T
              (OrderID               NUMBER(11,0)     NOT NULL,
               OrderDate             DATE DEFAULT SYSDATE,
               CustomerID            NUMBER(11,0),
CONSTRAINT Order_PK PRIMARY KEY (OrderID),
CONSTRAINT Order_FK FOREIGN KEY (CustomerID) REFERENCES Customer_T(CustomerID));
```

Foreign key of dependent table

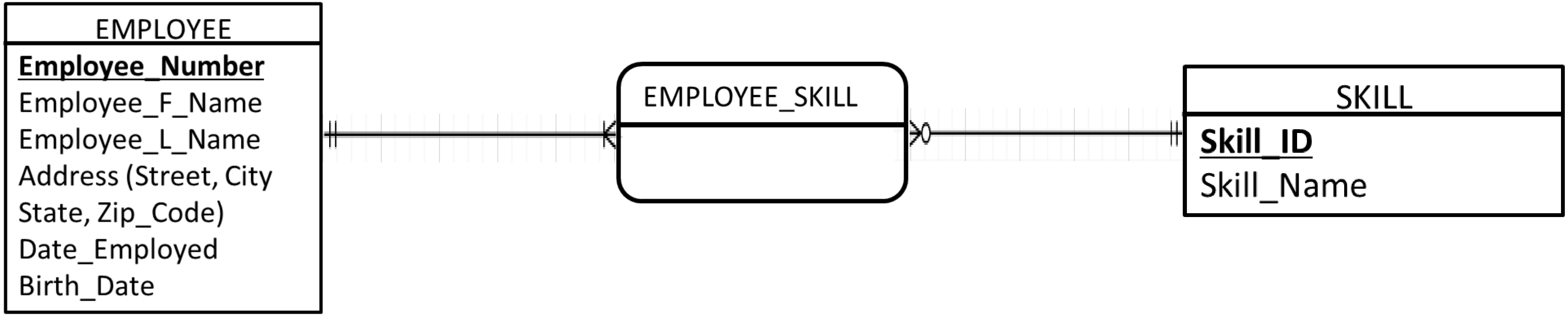
Class Activity 2.2:

Create table EMPLOYEE with **default value** for **Date_Employed** attribute.

BR1: The system needs to provide a unique number foreach employee, and collect employees' personal information including name, home address, date of birth, and employment date.

BR2: An employee needs to have experience with one or more general purpose programming languages including: Java, C/C++, C#.

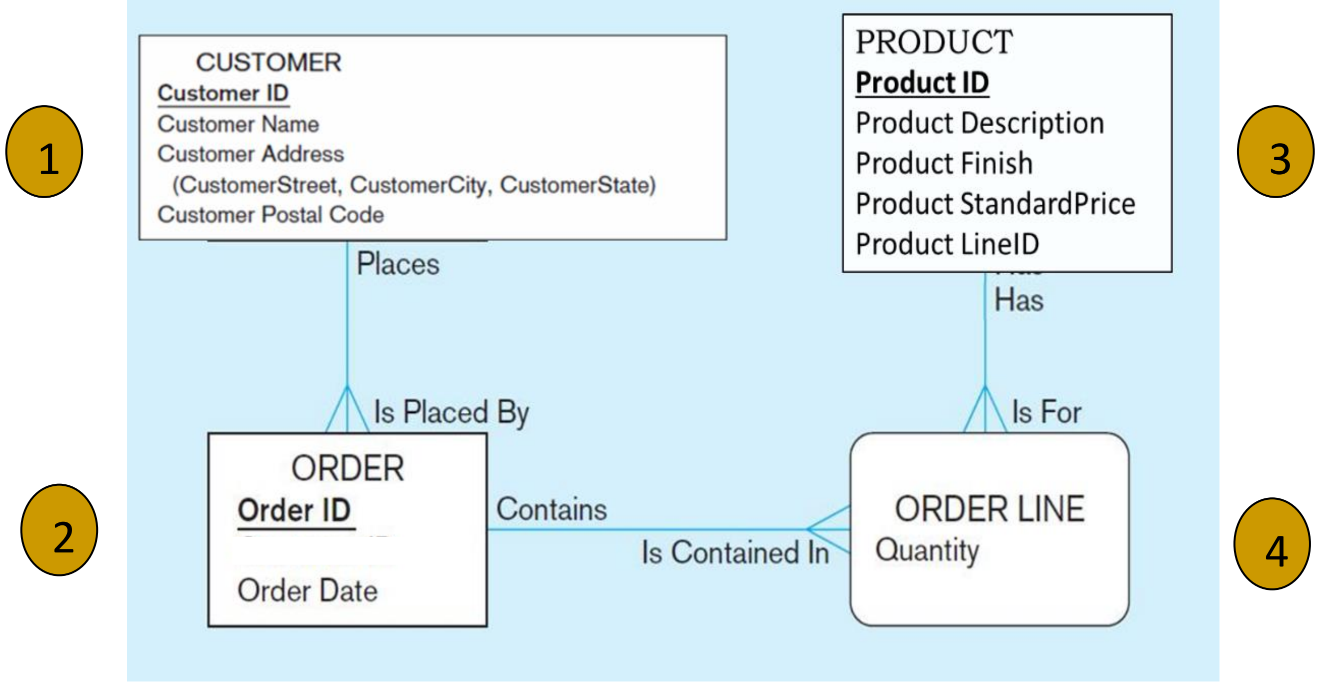
A1: Any skill can be taken by any number of employees.



The Create table order

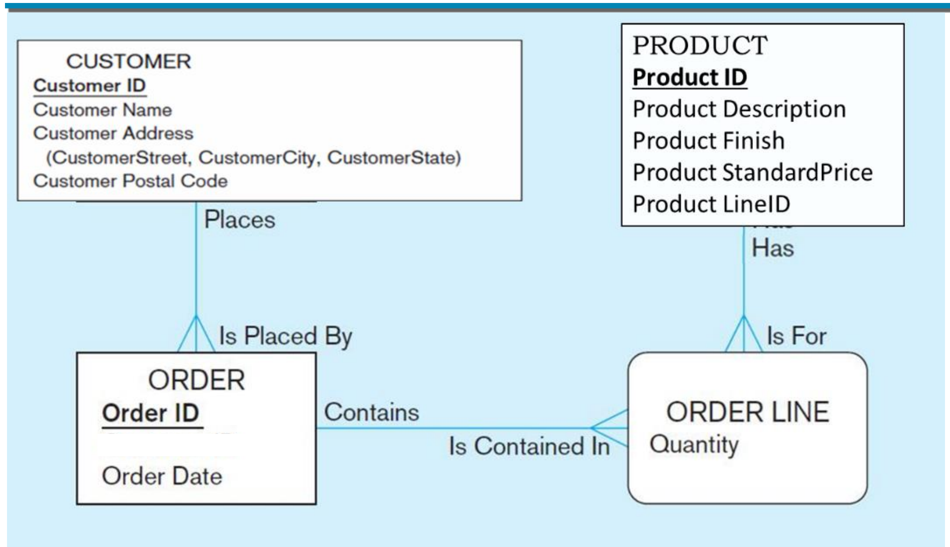
We have created CUSTOMER and ORDER tables.

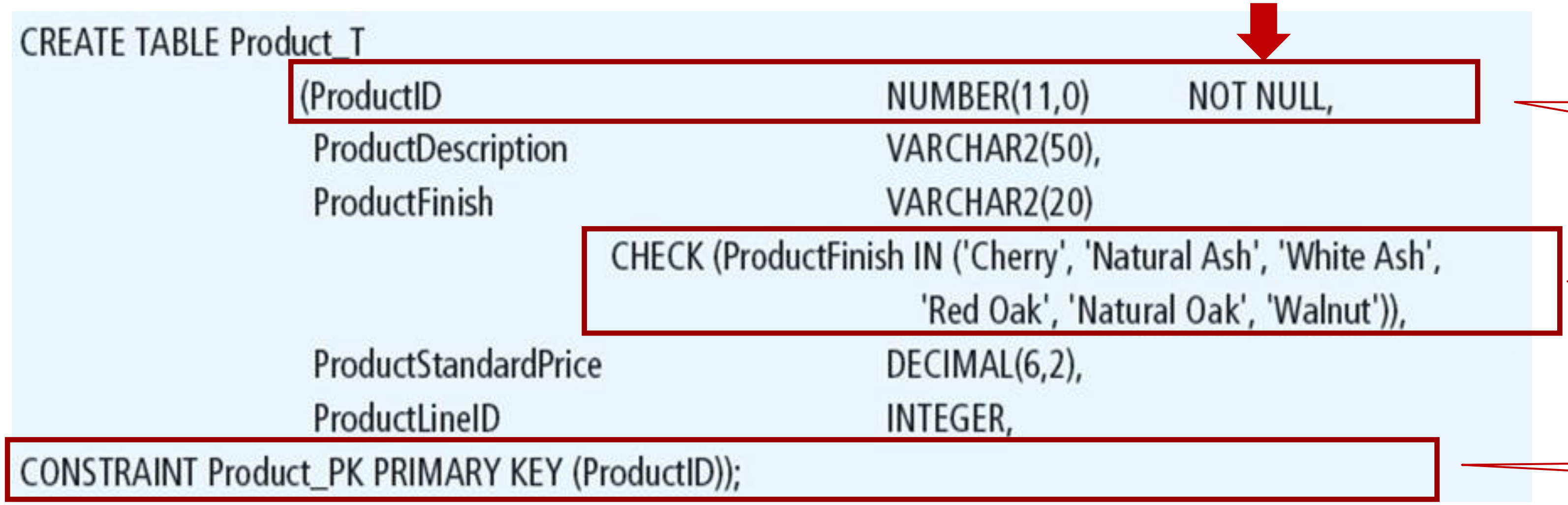
Can we create ORDERLINE table now?



Create PRODUCT Table

1. Identify data types for attributes
2. Identify columns that can and cannot be null
3. Identify columns that must be unique (candidate keys)
4. Identify primary key–foreign key mates
5. Determine default values
6. Identify constraints on columns (domain specifications)
7. Create the table and associated indexes





```
CREATE TABLE Product_T
        (ProductID                    NUMBER(11,0)      NOT NULL,
         ProductDescription           VARCHAR2(50),
         ProductFinish                VARCHAR2(20)
                   CHECK (ProductFinish IN ('Cherry', 'Natural Ash', 'White Ash',
                                 'Red Oak', 'Natural Oak', 'Walnut')),
         ProductStandardPrice         DECIMAL(6,2),
         ProductLineID                INTEGER,
CONSTRAINT Product_PK PRIMARY KEY (ProductID));
```

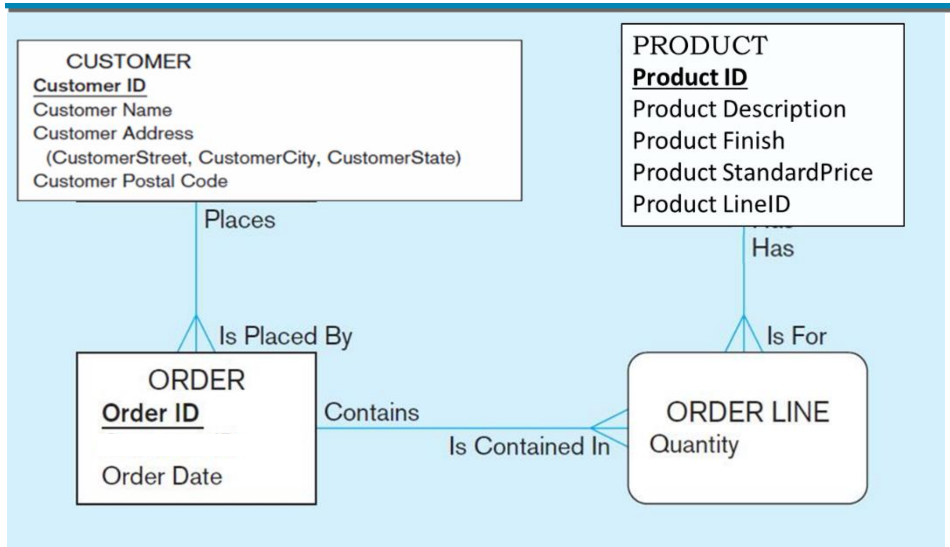
Primary keys can never have NULL values

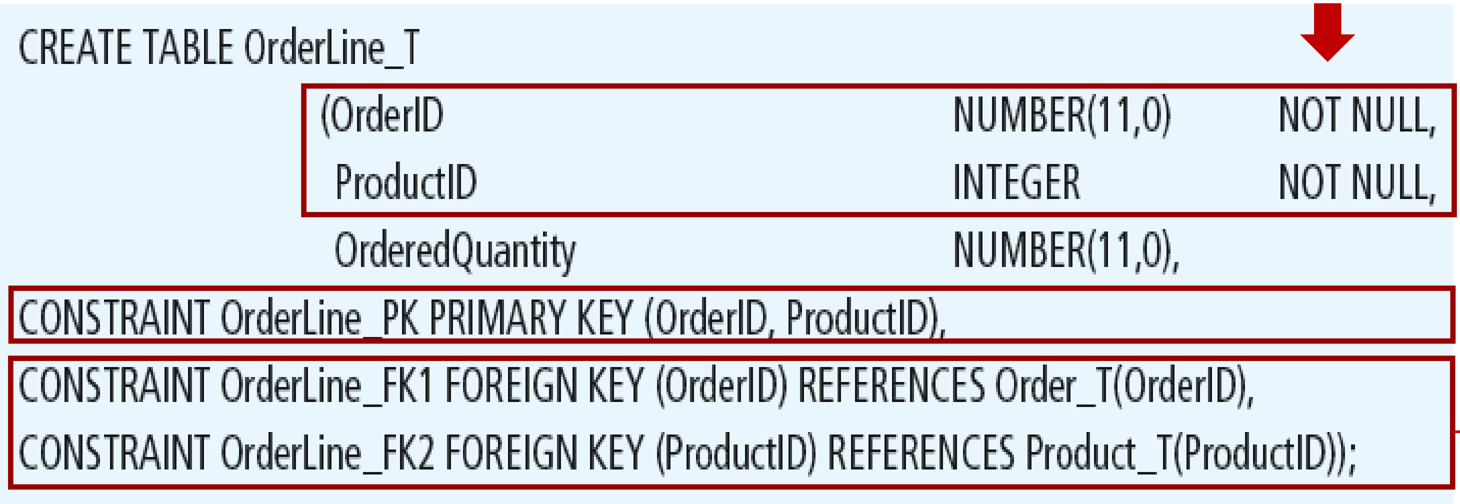
Domain constraint

PK Constraint

Create ORDERLINE Table

1. Identify data types for attributes
2. Identify columns that can and cannot be null
3. Identify columns that must be unique (candidate keys)
4. Identify primary key–foreign key mates
5. Determine default values
6. Identify constraints on columns (domain specifications)
7. Create the table and associated indexes





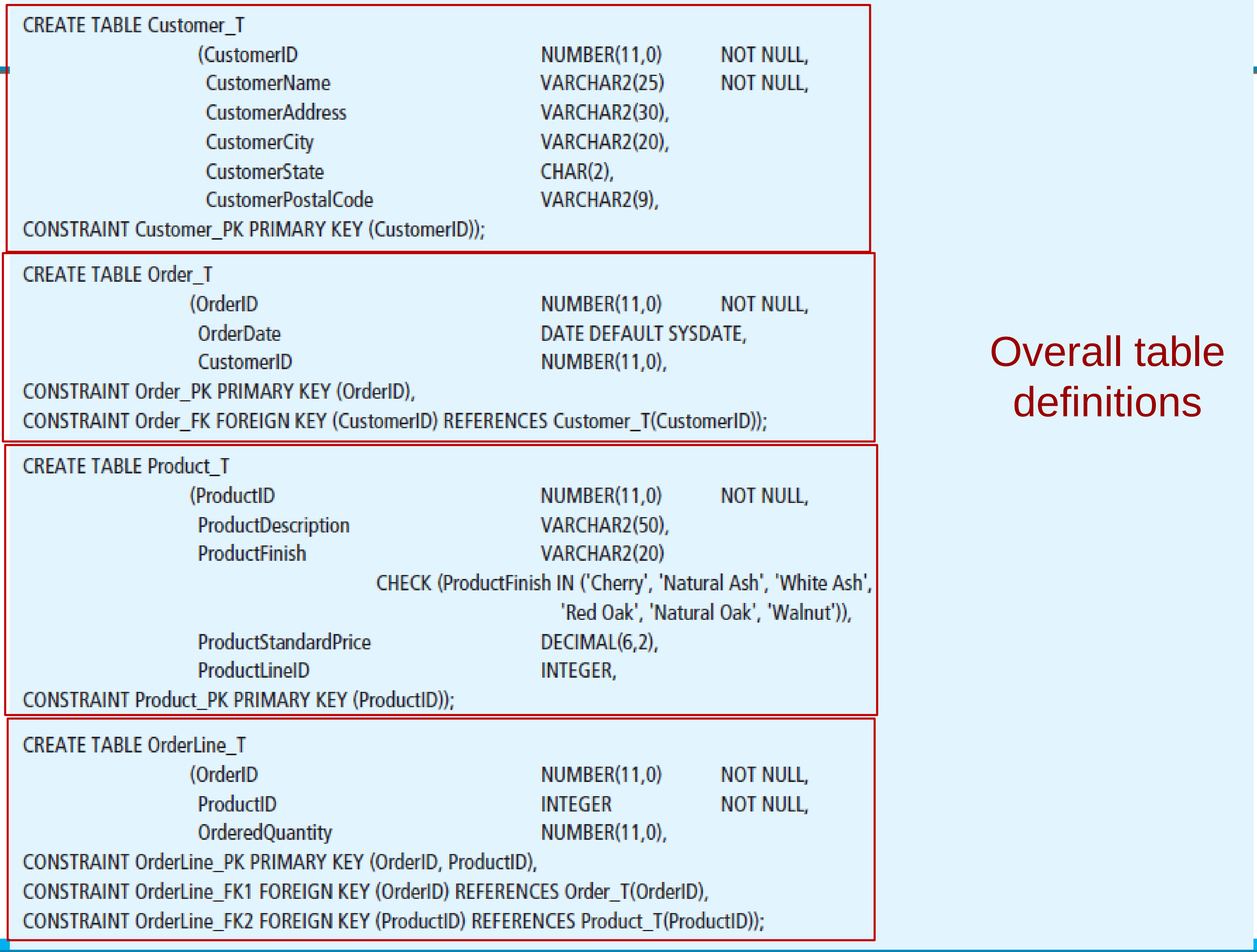
```
CREATE TABLE OrderLine_T
                (OrderID                    NUMBER(11,0)      NOT NULL,
                 ProductID                  INTEGER           NOT NULL,
                 OrderedQuantity            NUMBER(11,0),
CONSTRAINT OrderLine_PK PRIMARY KEY (OrderID, ProductID),
CONSTRAINT OrderLine_FK1 FOREIGN KEY (OrderID) REFERENCES Order_T(OrderID),
CONSTRAINT OrderLine_FK2 FOREIGN KEY (ProductID) REFERENCES Product_T(ProductID));
```

Composite PK can never have NULL values

PK Constraint

Foreign key of dependent table

SQL database definition commands for PVF Company (Figure 6-6)



```
CREATE TABLE Customer_T
                    (CustomerID                 NUMBER(11,0)       NOT NULL,
                     CustomerName               VARCHAR2(25)       NOT NULL,
                     CustomerAddress            VARCHAR2(30),
                     CustomerCity               VARCHAR2(20),
                     CustomerState              CHAR(2),
                     CustomerPostalCode         VARCHAR2(9),
CONSTRAINT Customer_PK PRIMARY KEY (CustomerID));

CREATE TABLE Order_T
                   (OrderID                     NUMBER(11,0)       NOT NULL,
                    OrderDate                   DATE DEFAULT SYSDATE,
                    CustomerID                  NUMBER(11,0),
CONSTRAINT Order_PK PRIMARY KEY (OrderID),
CONSTRAINT Order_FK FOREIGN KEY (CustomerID) REFERENCES Customer_T(CustomerID));

CREATE TABLE Product_T
                   (ProductID                   NUMBER(11,0)       NOT NULL,
                    ProductDescription          VARCHAR2(50),
                    ProductFinish               VARCHAR2(20)
                                    CHECK (ProductFinish IN ('Cherry', 'Natural Ash', 'White Ash',
                                                 'Red Oak', 'Natural Oak', 'Walnut')),
                    ProductStandardPrice        DECIMAL(6,2),
                    ProductLineID               INTEGER,
CONSTRAINT Product_PK PRIMARY KEY (ProductID));

CREATE TABLE OrderLine_T
                   (OrderID                     NUMBER(11,0)       NOT NULL,
                    ProductID                   INTEGER            NOT NULL,
                    OrderedQuantity             NUMBER(11,0),
CONSTRAINT OrderLine_PK PRIMARY KEY (OrderID, ProductID),
CONSTRAINT OrderLine_FK1 FOREIGN KEY (OrderID) REFERENCES Order_T(OrderID),
CONSTRAINT OrderLine_FK2 FOREIGN KEY (ProductID) REFERENCES Product_T(ProductID));
```

Overall table definitions

(Oracle 12c)

Class Activity 2.3: Create table EMPLOYEE.

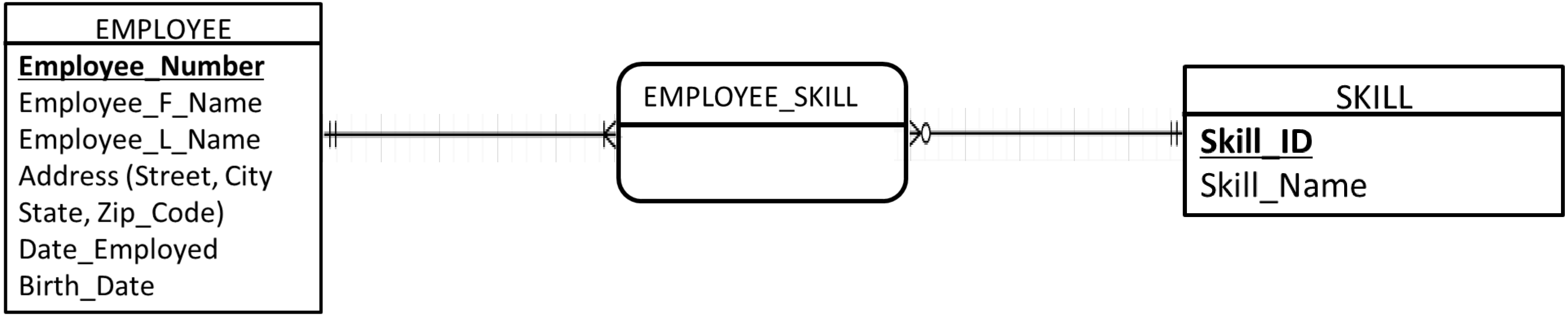
Notes:

- Just **NSW, VIC, WA, SA, ACT** and **NT** states can be inserted in **State column** of the EMPLOYEE table.
- Address in EMPLOYEE Table should be divided into number, street, etc.

BR1: The system needs to provide a unique number foreach employee, and collect employees’ personal information including name, home address, date of birth, and employment date.

BR2: An employee needs to have experience with one or more general purpose programming languages including: Java, C/C++, C#.

A1: Any skill can be taken by any number of employees.



EMPLOYEE

Employee_Number

Employee_F_Name

Employee_L_Name

Address (Street, City State, Zip_Code)

Date_Employed

Birth_Date

EMPLOYEE_SKILL

SKILL

Skill_ID

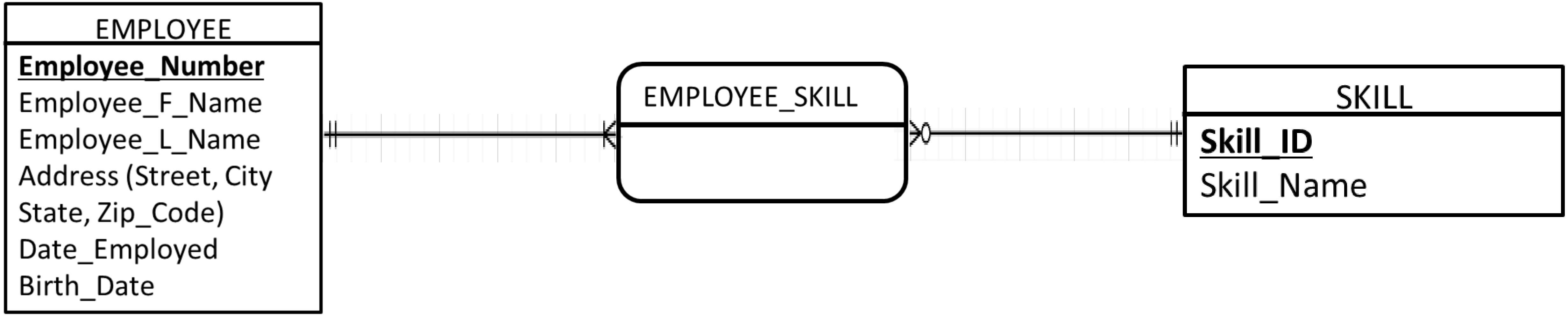
Skill_Name

Class Activity 2.4: Create table EMPLOYEE_SKILL.

BR1: The system needs to provide a unique number foreach employee, and collect employees' personal information including name, home address, date of birth, and employment date.

BR2: An employee needs to have experience with one or more general purpose programming languages including: Java, C/C++, C#.

A1: Any skill can be taken by any number of employees.



Summary of useful statement to use in Part D

- Data Integrity Controls
- Changing Tables
- Removing Tables
- Insert Statement
- Delete Statement
- Update Statement

Data Integrity Controls

For Example: The order of creating or dropping tables (PK/FK)

- Tables will not be dropped if there are other tables that depend on them. This means that if any table has a foreign key to the table being dropped, the drop will fail. Therefore, it makes a difference which order you drop the tables in.

Removing Tables

- DROP TABLE statement allows you to remove tables from your schema:

```
DROP TABLE CUSTOMER_T
```

Note: Tables will not be dropped if there are other tables that depend on them. This means that if any table has a foreign key to the table being dropped, the drop will fail. Therefore, it makes a difference which order you drop the tables in.

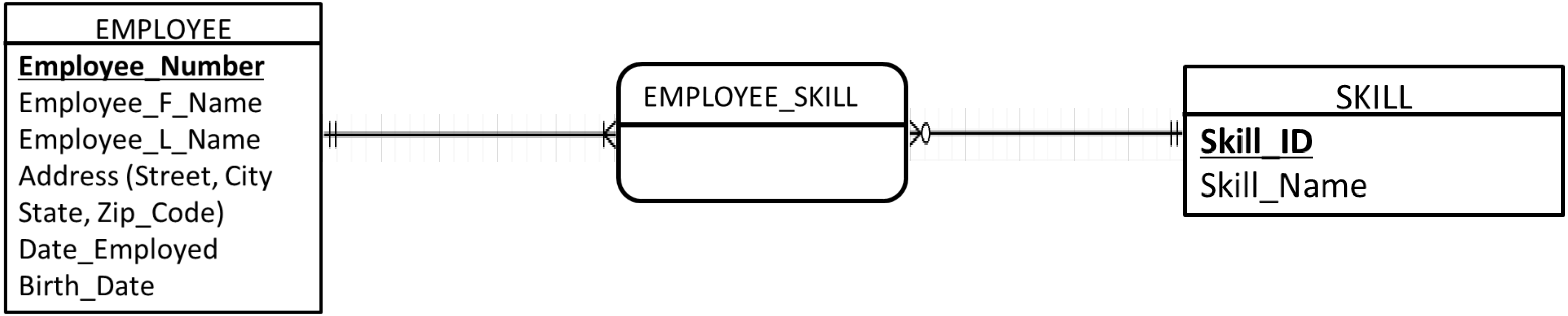
Class Activity 2.5:

Drop table EMPLOYEE_SKILL. What is the problem?

BR1: The system needs to provide a unique number foreach employee, and collect employees’ personal information including name, home address, date of birth, and employment date.

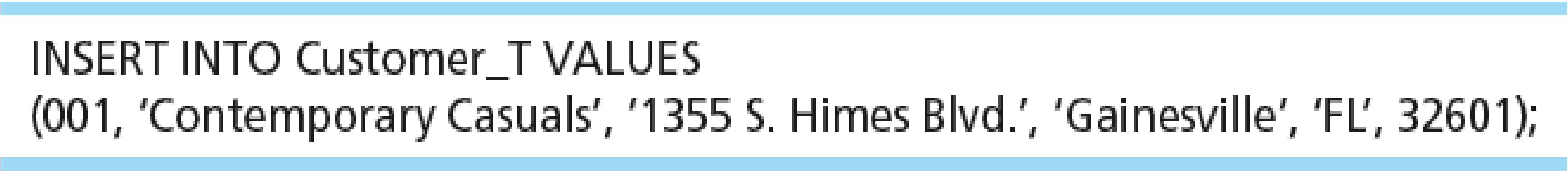
BR2: An employee needs to have experience with one or more general purpose programming languages including: Java, C/C++, C#.

A1: Any skill can be taken by any number of employees.



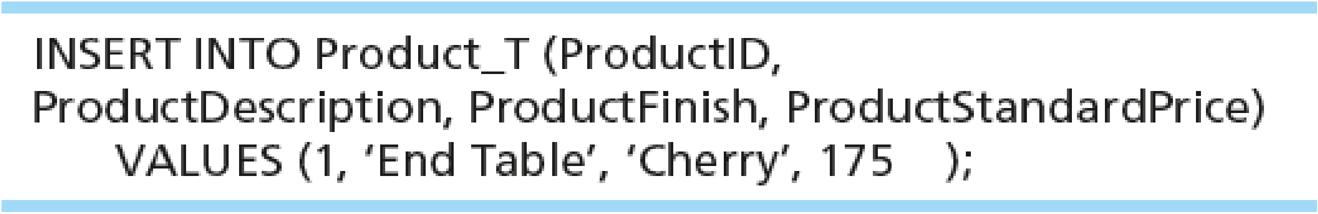
Insert Statement: Adds one or more rows to a table

- Inserting into a table



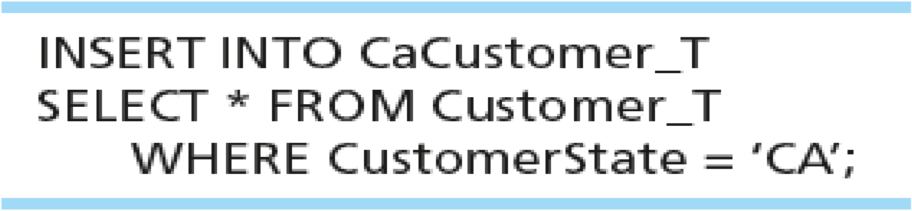
```
INSERT INTO Customer_T VALUES
(001, ‘Contemporary Casuals’, ‘1355 S. Himes Blvd.’, ‘Gainesville’, ‘FL’, 32601);
```

- Inserting a record that has some null attributes requires identifying the fields that actually get data



```
INSERT INTO Product_T (ProductID,
ProductDescription, ProductFinish, ProductStandardPrice)
    VALUES (1, ‘End Table’, ‘Cherry’, 175    );
```

- Inserting from another table (For your Information)



```
INSERT INTO CaCustomer_T
SELECT * FROM Customer_T
    WHERE CustomerState = ‘CA’;
```

Class Activity 2.6:

Insert **one row** into each table.

Note: Address in EMPLOYEE Table should be divided into number, street, etc.

EMPLOYEE

Employee_Number
Employee_F_Name
Employee_L_Name
Address (Street, City
State, Zip_Code)
Date_Employed
Birth_Date

EMPLOYEE-SKILL

Employee_Number
Skill_ID

SKILL

Skill_ID
Skill_Name

Employee_Number	Employee_F_Name	Employee_L_Name	Address	Date_Employed	Birth_Date
1123	Sara	Brown	UTS	1/1/2014	1/1/1985
1456	Jake	Cooper	32/50 …	5/8/2013	7/8/1990
7892	Fahimeh	Ramezani	12/97 …	2/3/2013	8/7/1987
8764	Ricky	Romanous	45/34 …	2/3/2015	4/3/1982

Employee_Number	Skill_ID
1123	A23
1123	B86
1456	C55
1456	A23
1456	C45

Skill_ID	Skill_Name
A23	Java
B86	C++
C55	C#
C45	Python

Delete Statement: Removes rows from a table

- Delete certain rows

```
DELETE FROM CUSTOMER_T
WHERE CUSTOMERSTATE = 'HI';
```

- Delete all rows

```
DELETE FROM CUSTOMER_T;
```

Remember, referential integrity rules will control whether a delete actually happens. The RESTRICT, CASCADE, and SET NULL constraints will determine how to handle the orders for a deleted customer.

Class Activity 2.7:

Delete **one row** from **EMPLOYEE_SKILL** table. What is the problem?

EMPLOYEE

Employee_Number
Employee_F_Name
Employee_L_Name
Address (Street, City
State, Zip_Code)
Date_Employed
Birth_Date

EMPLOYEE-SKILL

Employee_Number
Skill_ID

SKILL

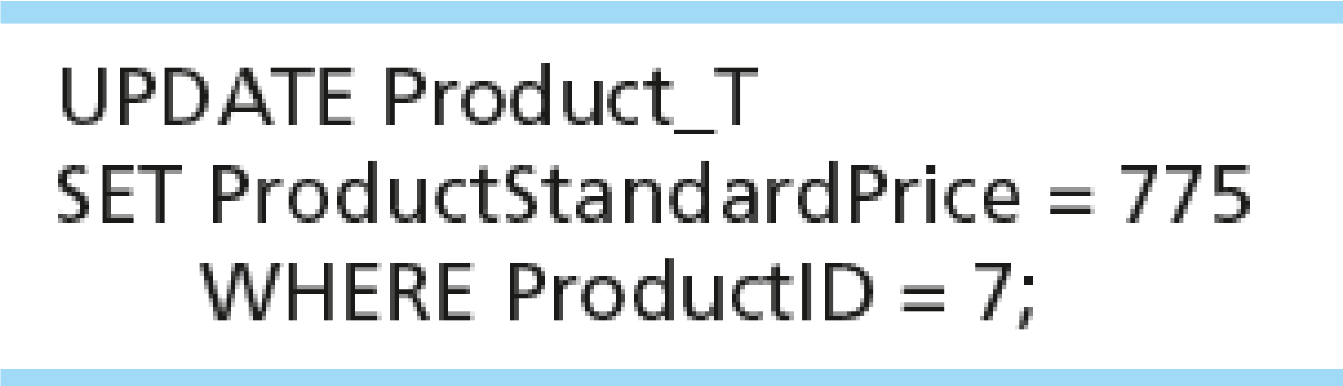
Skill_ID
Skill_Name

Employee_Number	Employee_F_Name	Employee_L_Name	Address	Date_Employed	Birth_Date
1123	Sara	Brown	UTS	1/1/2014	1/1/1985
1456	Jake	Cooper	32/50 …	5/8/2013	7/8/1990
7892	Fahimeh	Ramezani	12/97 …	2/3/2013	8/7/1987
8764	Ricky	Romanous	45/34 …	2/3/2015	4/3/1982

Employee_Number	Skill_ID
1123	A23
1123	B86
1456	C55
1456	A23
1456	C45

Skill_ID	Skill_Name
A23	Java
B86	C++
C55	C#
C45	Python

Update Statement: Modifies data in existing rows



```
UPDATE Product_T
SET ProductStandardPrice = 775
    WHERE ProductID = 7;
```

For this UPDATE, we know that it will affect only one record in the table. How do we know this?

Answer: Because ProductID is the primary key, which must be unique. So, there can be only one product with ProductID = 7.

However, many times updates and deletes affect many records. For example,

DELETE FROM CUSTOMER_T WHERE CUSTOMERSTATE = 'HI’;
affects all customers from Hawaii.

Class Activity 2.8:

Update **Employee_F_Name for Employee ID 1123 to Sam.**

EMPLOYEE

Employee_Number
Employee_F_Name
Employee_L_Name
Address (Street, City
State, Zip_Code)
Date_Employed
Birth_Date

EMPLOYEE-SKILL

Employee_Number
Skill_ID

SKILL

Skill_ID
Skill_Name

Employee_Number	Employee_F_Name	Employee_L_Name	Address	Date_Employed	Birth_Date
1123	Sara	Brown	UTS	1/1/2014	1/1/1985
1456	Jake	Cooper	32/50 ...	5/8/2013	7/8/1990
7892	Fahimeh	Ramezani	12/97 ...	2/3/2013	8/7/1987
8764	Ricky	Romanous	45/34 ...	2/3/2015	4/3/1982

Employee_Number	Skill_ID
1123	A23
1123	B86
1456	C55
1456	A23
1456	C45

Skill_ID	Skill_Name
A23	Java
B86	C++
C55	C#
C45	Python

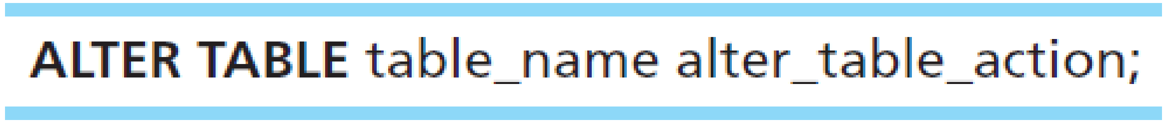
Changing Tables (For your Information)

- The ALTER command will be done after tables have already been created. For example, if you have an existing database, even one with actual data in it, you can modify tables by adding or changing columns, removing columns adding constraints, etc. If data in the tables violate the constraints, you will be prevented from setting these constraints until after changing the data.

- So, whereas CREATE TABLE is mostly a process that takes place during implementation, ALTER TABLE often takes place during maintenance.

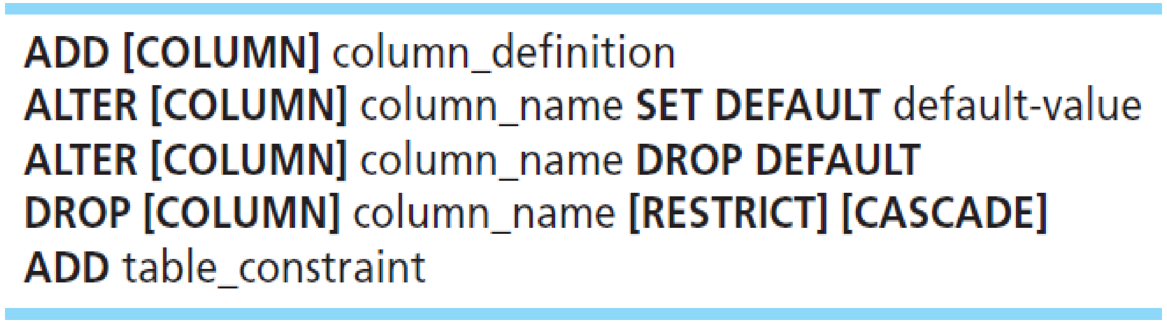
Changing Tables (For your Information)

- ALTER TABLE statement allows you to change column specifications:



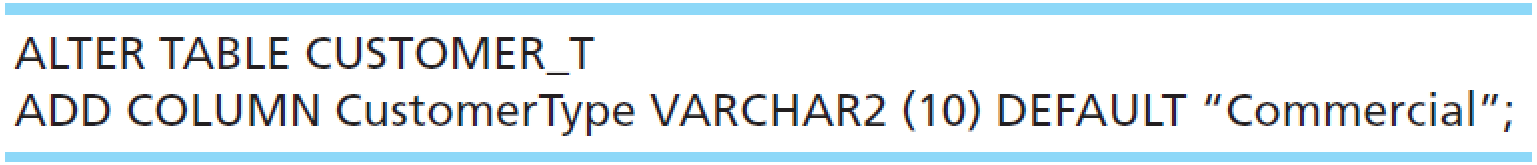
```
ALTER TABLE table_name alter_table_action;
```

- Table Actions:



```
ADD [COLUMN] column_definition
ALTER [COLUMN] column_name SET DEFAULT default-value
ALTER [COLUMN] column_name DROP DEFAULT
DROP [COLUMN] column_name [RESTRICT] [CASCADE]
ADD table_constraint
```

- Example (adding a new column with a default value):



```
ALTER TABLE CUSTOMER_T
ADD COLUMN CustomerType VARCHAR2 (10) DEFAULT “Commercial”;
```

Creating Tables with Identity Columns (For your Information)

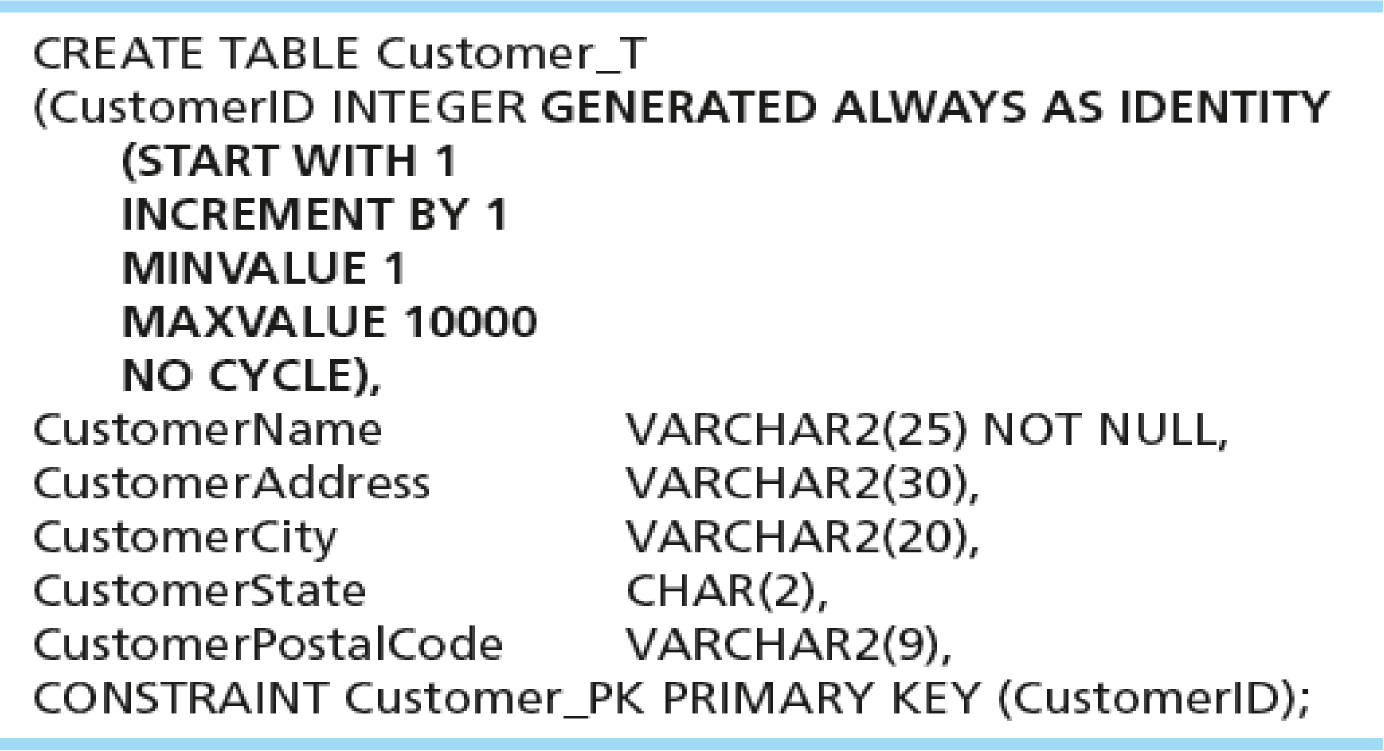
Identity columns are columns whose value automatically increment with each new INSERT.

So, an INSERT statement does not explicitly give a value for an identity column; this is handled automatically.

Often primary keys are identity columns, but not always.

Creating Tables with Identity Columns (For your Information)

Introduced with SQL:2008



```
CREATE TABLE Customer_T
(CustomerID INTEGER GENERATED ALWAYS AS IDENTITY
      (START WITH 1
      INCREMENT BY 1
      MINVALUE 1
      MAXVALUE 10000
      NO CYCLE),
CustomerName              VARCHAR2(25) NOT NULL,
CustomerAddress           VARCHAR2(30),
CustomerCity              VARCHAR2(20),
CustomerState             CHAR(2),
CustomerPostalCode        VARCHAR2(9),
CONSTRAINT Customer_PK PRIMARY KEY (CustomerID);
```

Inserting into a table does not require explicit customer ID entry or field list.

```
INSERT INTO CUSTOMER_T VALUES ( 'Contemporary Casuals', '1355 S. Himes Blvd.', 'Gainesville', 'FL', 32601);
```

More Constraints to set (For your Information)

Unique Constraint

Imagine that we have a BR like this:

> **BR:** There should not be more than one employee with the same first and last name in this company.

Therefore, when we create EMPLOYEE table, we need to keep the **combination of same first and last name unique** for each employee.

```
CREATE TABLE Employee
(Employee_Number       int        NOT NULL,
Employee_F_Name      VARCHAR(50),
Employee_L_Name    VARCHAR(50),
CONSTRAINT Employee_PK PRIMARY KEY (Employee_Number),
CONSTRAINT Employee_UQ UNIQUE (Employee_F_Name , Employee_L_Name)
);
```

Note: some columns of the EMPLOYEE table are not included in this sample

Check the Unique Constraint in previous sample:

```
insert into Employee_F
values
    (11, 'Sara','Brown'),
    (12, 'Sara','Brown’);
```

What is the error? Why?

Summary

- Create tables
- Determined the data types
- Set the not null constraint --> identifier and required attributes
- Set the domain constraint (Check the value of a column to be in a set of values) → Optional
- Set default values (date ...) → Optional
- Determine the PK/FK relations
- **Set PK constraint**
- **Set FK constraint**
- Determine composite PK
- The FK data type should match the related PK data type
- The order if creating tables should be considered

Next Lectures …

Week 3. Modeling Relationships:

3.1. Relationship Types vs. Relationship Instances

3.2. Degree of Relationships

3.3. Cardinality of Relationships

3.4. Multiple Relationships Between Entities

3.5. Multivalued Attributes Can be Represented as Relationships

3.6. Relationships Can Have Attributes

3.7. Associative Entity– Combination of Relationship and Entity

3.8. Identifying Relationship – Weak and Strong Entities

Notations: Basic E-R Notation