

104KM Enterprise Information System

Topic: Relational Model and Relational Database Design – Part 1

Learning outcomes for today

- Relational Model – Terminologies
- What is SQL?
- SQL Commands
- Examples

What is a Database?

- A **Database** is an organized collection of data, typically stored in electronic format.
 - It allows you to input, manage, organize and retrieve data quickly.
 - Traditional databases are organized by records (rows), fields (columns) stored in tables which are stored in database files.
 - Is a **file** used to store information.
- A database **table** is a collection of rows and columns that is used to organize information about a single topic or object. Each row within a table corresponds to a single record and contains several different attributes that describe the row.
- A database **table** is the most common and simplest form of data storage in a relational database.

Solution from Tutorial Activity 2 – Build a Database

EMPLOYEE DATABASE

Employee_Personal_Details

Employee_Id	Employee_Name	Employee_Addr
E110	Ali	12A, Plymouth Road, Devon
E111	Mary	423, Tudor Avenue, Coventry
E112	John	3, Merlin Close, London
E113	Peter	10, Jardine Road, Manchester

Employee Job

Employee_Id	Job_Role	Department_Name	Salary
E110	Accountant	Finance Department	40 000
E111	HR Manager	HR Department	30 000
E112	DB Administrator	IT Department	60 000
E113	Programmer	IT Department	50 000

Employee_Project

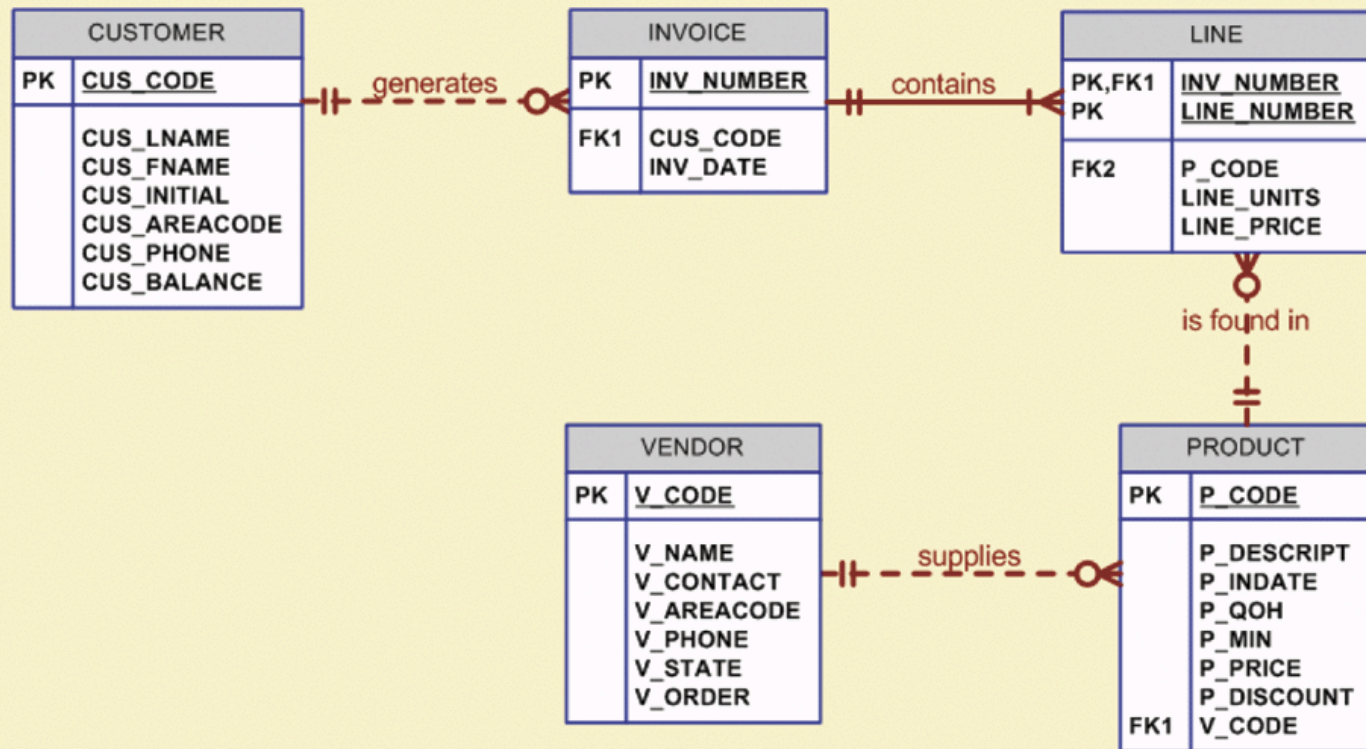
Employee_Id	Project_Name	Project_Start_Date	Project_End_Date	Total Hours
E110	Project_XXX	6/1/2013	6/4/2013	420
E111	Project_YYY	10/5/2013	10/6/2013	140
E112	Project_ZZZ	4/10/2014	20/12/2014	300
E113	Project_QQQ	1/8/2015	31/8/2015	100

What is a Relational Database Management System (RDBMS)?

- Database Management System = DBMS
- Relational DBMS = RDBMS
- A software system designed to allow the definition, creation, querying and updating of data stored in a relational database.
- *Examples: Oracle, DB2, MySQL, Microsoft SQL Server*

Example - The Relational Database Model

FIGURE 7.1
The database model



Question

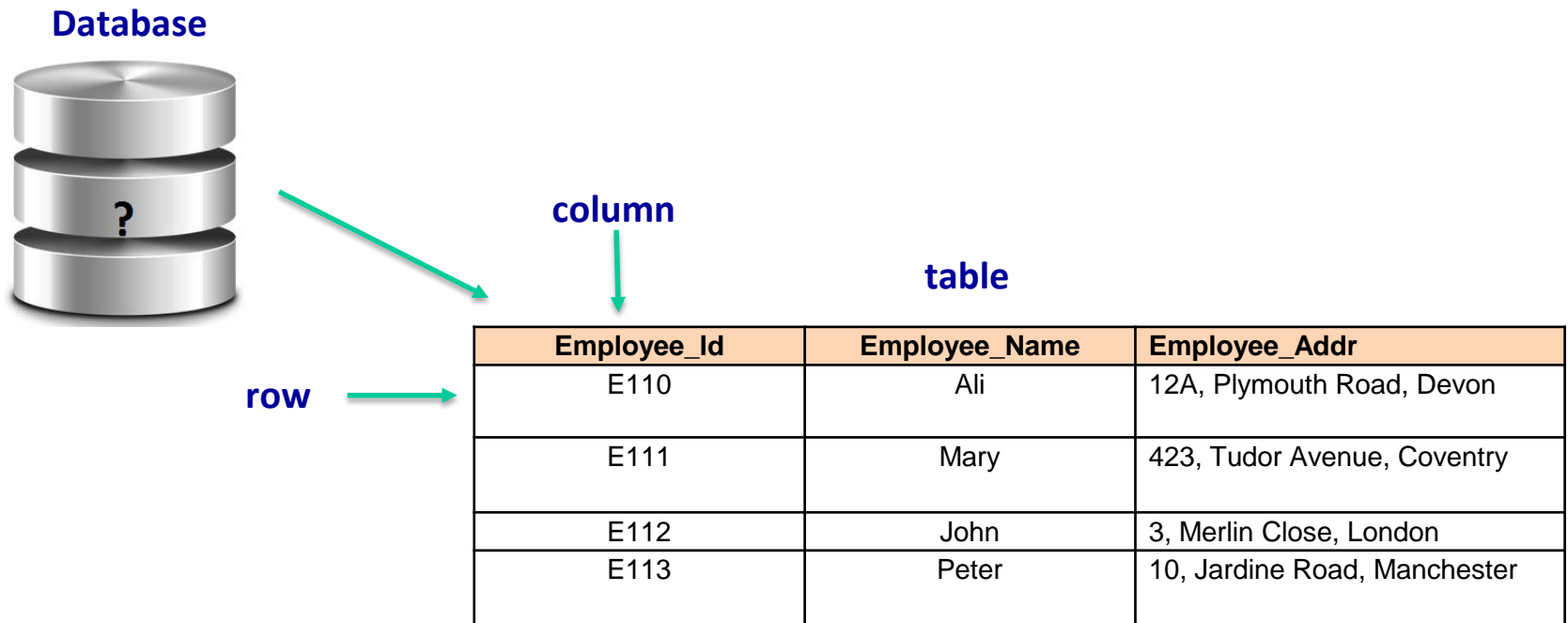
Where are RDBMS used?

- *Backend for traditional database application*
- *Backend for large Websites*
- *Backend for Web Services*

Terminologies Related to Relational Model

1. **Relation**: A relation is defined as a table with columns and rows.

According to E.F. Codd, data can be stored in form of a two-dimensional table.



2. **Tuple**: It is a row of a relation.

Terminologies Related to Relational Model

3. **Attributes**: It is defined as a named column of a relation.

Example of attributes:

➔ *Employee_id*

➔ *Employee_Name*

➔ *Employee_Addr*

4. **Degree**: The degree of a relation is defined as the number of attributes it contains.

For example,

if a table have 3 attributes, then it's degree is 3.

Objectives of SQL

Ideally, a database language allows programmer to:

- ▶ create the database and relation structures
- ▶ perform basic data management tasks, such as the insertion, modification and deletion of data from the relations
- ▶ perform both simple and complex queries
(as Query engine)

What is SQL?

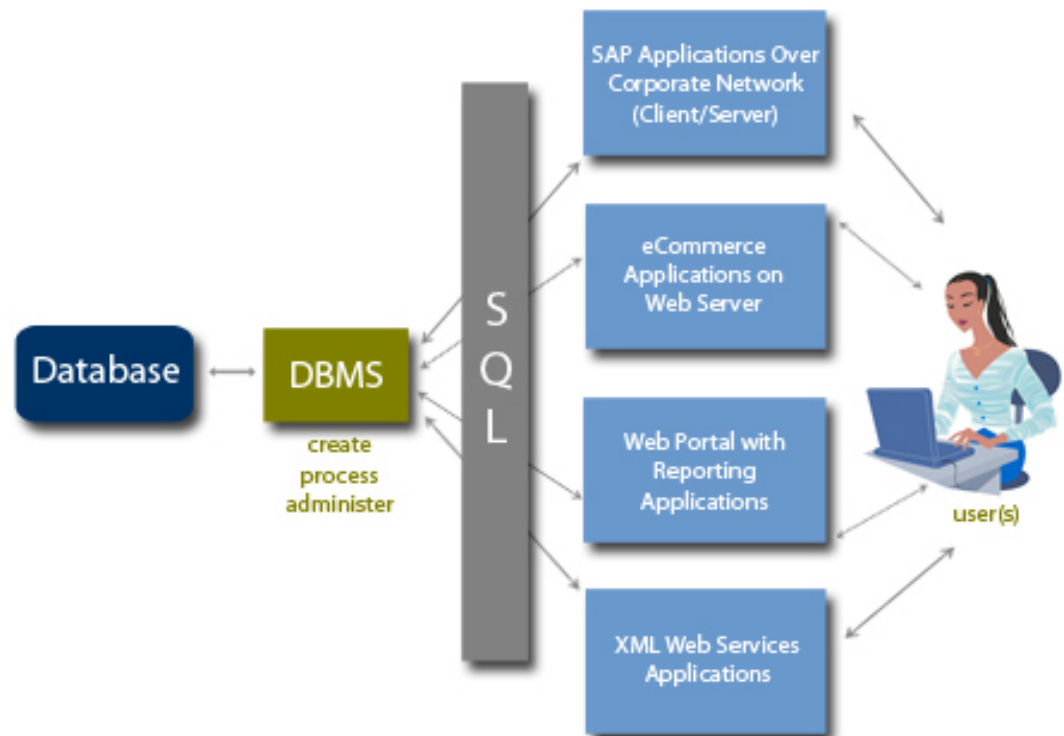
- ▶ SQL - Structured Query Language (*often pronounced as sequel*)
- ▶ SQL is a special-purpose programming language used for managing data held in a Relational Database Management System (RDBMS).
- ▶ Overall, SQL includes:
 - ▶ creating tables
 - ▶ insert data, query, update, delete
 - ▶ data access control

Types of SQL

- ▶ Microsoft SQL Server using T-SQL
- ▶ **Oracle using PL/SQL** (We will be using this one!)
(PL/SQL is a procedural language design to embrace SQL statements within its syntax.)
- ▶ Microsoft Access version of SQL

Why do we need SQL?

- ▶ To model the relational database
- ▶ To manage the data within the database
- ▶ To query the database



SQL Functions

SQL functions fit into **two** broad categories:

Data Definition Language (DDL)

SQL includes commands to:

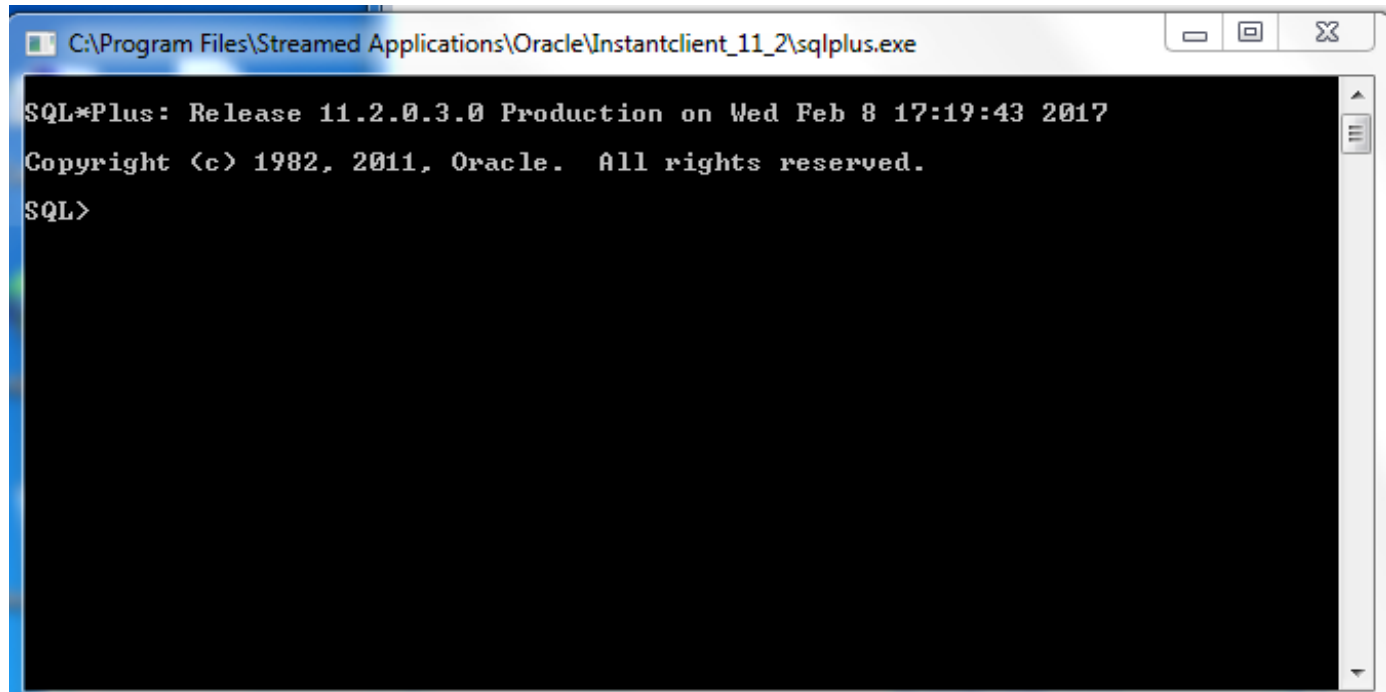
- Create database objects such as tables, indexes and views
- Define access rights to those database objects

Data Manipulation Language (DML)

- Includes commands to insert, update, delete, and retrieve data within database tables.

Oracle SQL+

Below is the screenshot of an Oracle SQL+ you will be using in the lab:



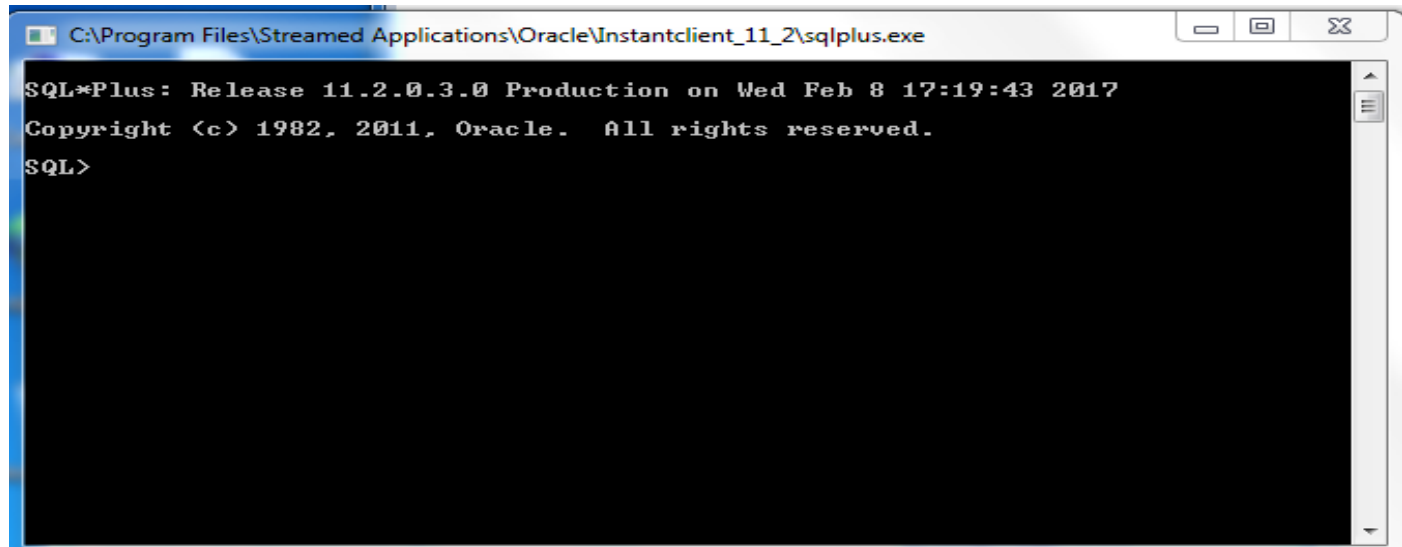
The screenshot shows a Windows command prompt window titled "C:\Program Files\Streamed Applications\Oracle\Instantclient_11_2\sqlplus.exe". The window has standard Windows window controls (minimize, maximize, close) in the top right corner. The main area of the window is black with white text. The text displayed is:

```
SQL*Plus: Release 11.2.0.3.0 Production on Wed Feb 8 17:19:43 2017  
Copyright (c) 1982, 2011, Oracle. All rights reserved.  
SQL>
```

Oracle SQL+

Step 1: Launch Oracle 11g with SQL+ from AppsAnywhere

Step 2: The following screen will appear:

A screenshot of a Windows command window titled "C:\Program Files\Streamed Applications\Oracle\Instantclient_11_2\sqlplus.exe". The window has a black background with white text. The text displayed is: "SQL*Plus: Release 11.2.0.3.0 Production on Wed Feb 8 17:19:43 2017", "Copyright (c) 1982, 2011, Oracle. All rights reserved.", and "SQL>".

```
C:\Program Files\Streamed Applications\Oracle\Instantclient_11_2\sqlplus.exe

SQL*Plus: Release 11.2.0.3.0 Production on Wed Feb 8 17:19:43 2017
Copyright (c) 1982, 2011, Oracle. All rights reserved.
SQL>
```

Wait until the SQL prompt

“SQL> ” is displayed, and then typed the following:

connect /@acal

SQL> quit <to exit>

Why Oracle SQL+?

- ▶ Oracle is used throughout the world.
- ▶ Command-line.
- ▶ Next year, you might explore more graphical packages that offer you tool to aid development of SQL statements.
- ▶ But now, **you need to understand how to develop SQL statements and queries.**

The Rules

- ▶ Objects, identifiers, etc. identified by names
- ▶ Names not case-sensitive
- ▶ Naming Rules:
 - ▶ must start with a letter
 - ▶ valid A-Z, a-z, 0-9, _ (\$,# not encouraged)

Rules

- ▶ Keywords cannot be split across lines
- ▶ Keywords normally written in uppercase
- ▶ Clauses are usually placed on separate lines
to improve readability and for ease of editing
- ▶ In SQL+, each statement is terminated with a “;”

Data Types

► String types

CHAR(n) – fixed-length character data, n characters long

Maximum length – 2000bytes

VARCHAR2(n) – variable length character data

Maximum 4000 bytes

LONG – variable-length character data, up to 4GB.

Max 1 per table

► **DATE** - 2-digit number for Day, 3 letters for month,

2 – digit number for the year

e.g. 7th January 2014 stored as 07-jan-14

More Data Types

► Numeric types

NUMBER(L,R) – general purpose numeric data type

L – max number of significant decimal digits or the precision

R – number of digits from the decimal point to the least significant digit or the scale

NUMBER(5,2) – max value 999.99

NUMBER(5) – max value 99999

► **INTEGER** – signed integer

Common SQL Data Types

TABLE
7.4

Some Common SQL Data Types

DATA TYPE	FORMAT	COMMENTS
Numeric	NUMBER(L,D)	The declaration NUMBER(7,2) indicates numbers that will be stored with two decimal places and may be up to six digits long, including the sign and the decimal place. Examples: 12.32, -134.99.
	INTEGER	May be abbreviated as INT. Integers are (whole) counting numbers, so they cannot be used if you want to store numbers that require decimal places.
	SMALLINT	Like INTEGER, but limited to integer values up to six digits. If your integer values are relatively small, use SMALLINT instead of INT.
	DECIMAL(L,D)	Like the NUMBER specification, but the storage length is a <i>minimum</i> specification. That is, greater lengths are acceptable, but smaller ones are not. DECIMAL(9,2), DECIMAL(9), and DECIMAL are all acceptable.
Character	CHAR(L)	Fixed-length character data for up to 255 characters. If you store strings that are not as long as the CHAR parameter value, the remaining spaces are left unused. Therefore, if you specify CHAR(25), strings such as "Smith" and "Katzenjammer" are each stored as 25 characters. However, a U.S. area code is always three digits long, so CHAR(3) would be appropriate if you wanted to store such codes.
	VARCHAR(L) or VARCHAR2(L)	Variable-length character data. The designation VARCHAR2(25) will let you store characters up to 25 characters long. However, VARCHAR will not leave unused spaces. Oracle users may use VARCHAR2 as well as VARCHAR.
Date	DATE	Stores dates in the Julian date format.

Data Definition Commands

- ▶ Concerned with the structure of the database. It defines the structure of database and controlling access to data.
- ▶ Example - Command statements:

CREATE

DROP

ALTER

COMMIT

TABLES

CREATE TABLE statement can be used to create a table.

Tables contain columns and constraints, rules to which data must conform.

Specify:

- name of the table
- names of column/attributes
- data type of each column/attribute
- size of each column/attribute

CREATE the table

CREATE TABLE Mytable

(id NUMBER(2),

name VARCHAR2(9));

SQL Constraints

The aim is to prevent invalid data in Databases. Commonly used constraints available in SQL are:

- NOT NULL Constraints: column must contain a value
- PRIMARY KEY Constraints: Uniquely identified each rows/records in a database table
- FOREIGN KEY: Uniquely identified a rows/records in another database table

Key in a table

- » A **key** in a table is a field or group of fields that creates identity, makes relationship with another table and/or make a table more efficient.

Primary Key & Foreign

- » A **Primary key** is a key that uniquely identifies a row in each table. It is normally denoted with its first two letters, namely, **PK**.
- » A **Foreign key** is a key borrowed from another related table (that's why its foreign) in order to make the relationship between two tables. It is normally denoted with its first two letters, namely, **FK**.

Example – Primary Key

Rep

RepNum	LastName	FirstName	Street	City	State	Zip	Commission	Rate
20	Kaiser	Valerie	624 Randall	Grove	FL	33321	\$20,542.50	0.05
35	Hull	Richard	532 Jackson	Sheldon	FL	33553	\$39,216.00	0.07
65	Perez	Juan	1626 Taylor	Fillmore	FL	33336	\$23,487.00	0.05



Repnum uniquely identifies the *Rep* table and is the primary key of this table.

Example - Primary Key

Customer

CustomerNum	CustomerName	Street	City	State	Zip	Balance	CreditLimit	RepNum
148	Al's Appliance and Sport	2837 Greenway	Fillmore	FL	33336	\$6,550.00	\$7,500.00	20
282	Brookings Direct	3827 Devon	Grove	FL	33321	\$431.50	\$10,000.00	35
356	Ferguson's	382 Wildwood	Northfield	FL	33146	\$5,785.00	\$7,500.00	65
408	The Everything Shop	1828 Raven	Crystal	FL	33503	\$5,285.25	\$5,000.00	35
462	Bargains Galore	3829 Central	Grove	FL	33321	\$3,412.00	\$10,000.00	65
524	Kline's	838 Ridgeland	Fillmore	FL	33336	\$12,762.00	\$15,000.00	20
608	Johnson's Department Store	372 Oxford	Sheldon	FL	33553	\$2,106.00	\$10,000.00	65
687	Lee's Sport and Appliance	282 Evergreen	Altonville	FL	32543	\$2,851.00	\$5,000.00	35
725	Deerfield's Four Seasons	282 Columbia	Sheldon	FL	33553	\$248.00	\$7,500.00	35
842	All Season	28 Lakeview	Grove	FL	33321	\$8,221.00	\$7,500.00	20



Customernum uniquely identifies the *Customer* table and is the primary key of this table.

Example – Foreign Key

Rep

RepNum	LastName	FirstName	Street	City	State	Zip	Commission	Rate
20	Kaiser	Valerie	624 Randall	Grove	FL	33321	\$20,542.50	0.05
35	Hull	Richard	532 Jackson	Sheldon	FL	33553	\$39,216.00	0.07
65	Perez	Juan	1626 Taylor	Fillmore	FL	33336	\$23,487.00	0.05

Repnum is a *Foreign key* borrowed from Rep table

Customer

CustomerNum	CustomerName	Street	City	State	Zip	Balance	CreditLimit	RepNum
148	Al's Appliance and Sport	2837 Greenway	Fillmore	FL	33336	\$6,550.00	\$7,500.00	20
282	Brookings Direct	3827 Devon	Grove	FL	33321	\$431.50	\$10,000.00	35
356	Ferguson's	382 Wildwood	Northfield	FL	33146	\$5,785.00	\$7,500.00	65
408	The Everything Shop	1828 Raven	Crystal	FL	33503	\$5,285.25	\$5,000.00	35
462	Bargains Galore	3829 Central	Grove	FL	33321	\$3,412.00	\$10,000.00	65
524	Kline's	838 Ridgeland	Fillmore	FL	33336	\$12,762.00	\$15,000.00	20
608	Johnson's Department Store	372 Oxford	Sheldon	FL	33553	\$2,106.00	\$10,000.00	65
687	Lee's Sport and Appliance	282 Evergreen	Altonville	FL	32543	\$2,851.00	\$5,000.00	35
725	Deerfield's Four Seasons	282 Columbia	Sheldon	FL	33553	\$248.00	\$7,500.00	35
842	All Season	28 Lakeview	Grove	FL	33321	\$8,221.00	\$7,500.00	20

Example

```
CREATE TABLE employees(  
  emp_id NUMBER(5) PRIMARY KEY,  
  emp_name VARCHAR2(15) NOT NULL,  
  dept_id NUMBER(4));
```

What about the Foreign Key?

```
CREATE TABLE employees(  
  emp_id NUMBER(5) PRIMARY KEY,  
  emp_name VARCHAR2(15) NOT NULL,  
  dept_id NUMBER(4),  
  CONSTRAINT emp_dept_fk  
  REFERENCES departments(dept_id));
```

OR

```
ALTER TABLE EMPLOYEES  
ADD CONSTRAINT DEPT_ID_FK FOREIGN KEY(DEPT_ID)  
REFERENCES DEPARTMENTS(DEPT_ID);
```


Altering a table

For changing the structure of a table:

- Adding a new column
- Modifying the definition of an existing column
- Deleting a column
- Defining a default value for a new column

Example:

ALTER TABLE employees

ADD address VARCHAR2(30);

Delete a Column

The rules:

- May or may not contain data
- Table must have at least one column remaining
- Primary key referenced by another column cannot be deleted without the cascade option

Example:

```
ALTER TABLE cust_table
```

```
DROP COLUMN cust_age;
```

Inserting the Data

Created the table? Yes, now insert the data.

Example:

```
INSERT INTO employees values (12345,'John Jack',5678);
```

COMMIT

- A COMMIT statement ends a transaction successfully.
- Make changes to a database permanent.
- Use a COMMIT statement to save your work.

Example:

COMMIT;

Drop a table

Drop Table:

- Removes table from the database

Example:

DROP TABLE tablename;

Summary - SQL Data Definition Commands

**TABLE
7.1**

SQL Data Definition Commands

COMMAND OR OPTION	DESCRIPTION
CREATE SCHEMA AUTHORIZATION	Creates a database schema
CREATE TABLE	Creates a new table in the user's database schema
NOT NULL	Ensures that a column will not have null values
UNIQUE	Ensures that a column will not have duplicate values
PRIMARY KEY	Defines a primary key for a table
FOREIGN KEY	Defines a foreign key for a table
DEFAULT	Defines a default value for a column (when no value is given)
CHECK	Constraint used to validate data in an attribute
CREATE INDEX	Creates an index for a table
CREATE VIEW	Creates a dynamic subset of rows/columns from one or more tables
ALTER TABLE	Modifies a table's definition (adds, modifies, or deletes attributes or constraints)
CREATE TABLE AS	Creates a new table based on a query in the user's database schema
DROP TABLE	Permanently deletes a table (and thus its data)
DROP INDEX	Permanently deletes an index
DROP VIEW	Permanently deletes a view

Summary - SQL Data Manipulation Commands

TABLE
7.2

SQL Data Manipulation Commands

COMMAND OR OPTION	DESCRIPTION
INSERT	Inserts row(s) into a table
SELECT	Selects attributes from rows in one or more tables or views
WHERE	Restricts the selection of rows based on a conditional expression
GROUP BY	Groups the selected rows based on one or more attributes
HAVING	Restricts the selection of grouped rows based on a condition
ORDER BY	Orders the selected rows based on one or more attributes
UPDATE	Modifies an attribute's values in one or more table's rows
DELETE	Deletes one or more rows from a table
COMMIT	Permanently saves data changes
ROLLBACK	Restores data to their original values

Summary - SQL Data Manipulation Commands

TABLE
7.2

SQL Data Manipulation Commands (continued)

COMMAND OR OPTION	DESCRIPTION
COMPARISON OPERATORS	
=, <, >, <=, >=, <>	Used in conditional expressions
LOGICAL OPERATORS	
AND/OR/NOT	Used in conditional expressions
SPECIAL OPERATORS	Used in conditional expressions
BETWEEN	Checks whether an attribute value is within a range
IS NULL	Checks whether an attribute value is null
LIKE	Checks whether an attribute value matches a given string pattern
IN	Checks whether an attribute value matches any value within a value list
EXISTS	Checks whether a subquery returns any rows
DISTINCT	Limits values to unique values
AGGREGATE FUNCTIONS	Used with SELECT to return mathematical summaries on columns
COUNT	Returns the number of rows with non-null values for a given column
MIN	Returns the minimum attribute value found in a given column
MAX	Returns the maximum attribute value found in a given column
SUM	Returns the sum of all values for a given column
AVG	Returns the average of all values for a given column

More Examples here

1. W3Schools.com

<http://www.w3schools.com/sql/>

2. <http://www.tutorialspoint.com/sql>