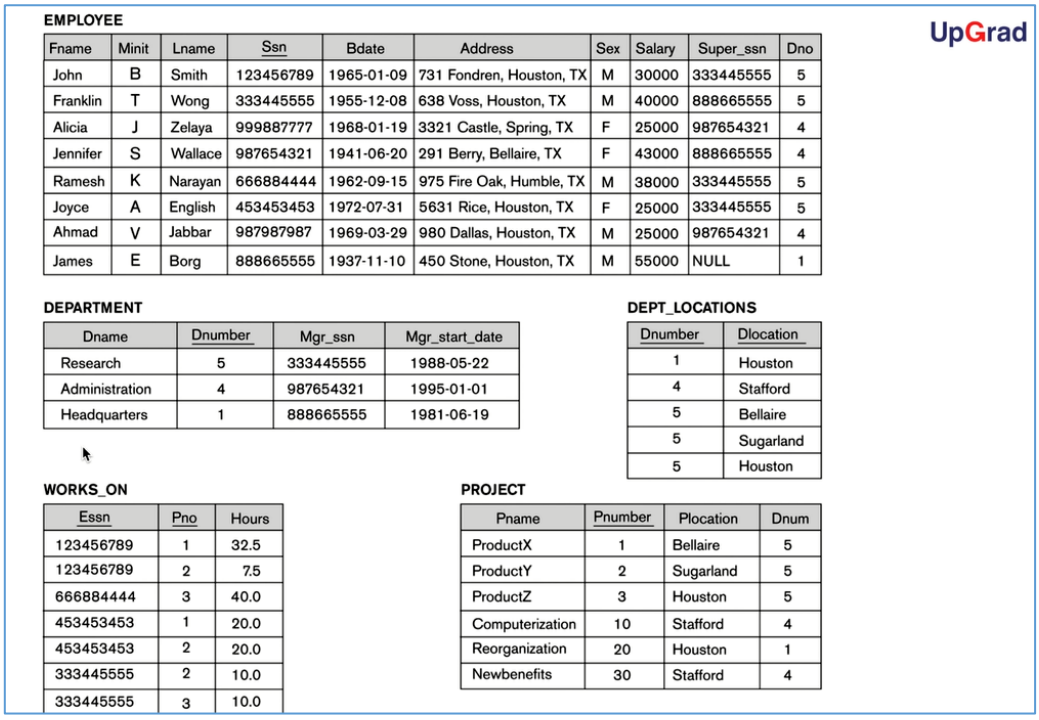
Module – Summary

Introduction to SQL

SQL (Structured Query Language) is the language used to communicate with databases. SQL is used nearly everywhere when it comes to data communication between a server and client. SQL has many implementations like SQLServer, MySQL, PostGreSQL etc. SQL queries are used to perform operations on a database like modifying, deleting, updating, subsetting etc.

Introduction to Relational Databases & MySQL

A relational database consists of various tables that are organized in a specific structure with tables sharing one or more ids common with one or more tables called the "relationship" between them.



EMPLOYEE

Fname	Minit	Lname	Ssn	Bdate	Address	Sex	Salary	Super_ssn	Dno
John	B	Smith	123456789	1965-01-09	731 Fondren, Houston, TX	M	30000	333445555	5
Franklin	T	Wong	333445555	1955-12-08	638 Voss, Houston, TX	M	40000	888665555	5
Alicia	J	Zelaya	999887777	1968-01-19	3321 Castle, Spring, TX	F	25000	987654321	4
Jennifer	S	Wallace	987654321	1941-06-20	291 Berry, Bellaire, TX	F	43000	888665555	4
Ramesh	K	Narayan	666884444	1962-09-15	975 Fire Oak, Humble, TX	M	38000	333445555	5
Joyce	A	English	453453453	1972-07-31	5631 Rice, Houston, TX	F	25000	333445555	5
Ahmad	V	Jabbar	987987987	1969-03-29	980 Dallas, Houston, TX	M	25000	987654321	4
James	E	Borg	888665555	1937-11-10	450 Stone, Houston, TX	M	55000	NULL	1

DEPARTMENT

Dname	Dnumber	Mgr_ssn	Mgr_start_date
Research	5	333445555	1988-05-22
Administration	4	987654321	1995-01-01
Headquarters	1	888665555	1981-06-19

DEPT_LOCATIONS

Dnumber	Dlocation
1	Houston
4	Stafford
5	Bellaire
5	Sugarland
5	Houston

WORKS_ON

Essn	Pno	Hours
123456789	1	32.5
123456789	2	7.5
666884444	3	40.0
453453453	1	20.0
453453453	2	20.0
333445555	2	10.0
333445555	3	10.0

PROJECT

Pname	Pnumber	Plocation	Dnum
ProductX	1	Bellaire	5
ProductY	2	Sugarland	5
ProductZ	3	Houston	5
Computerization	10	Stafford	4
Reorganization	20	Houston	1
Newbenefits	30	Stafford	4

Figure 1: Sample Database

A relational database contains information in various tables ranging in a few tables to thousands of tables. A row in a table is called a record and a column in a table is called a field.

In our session, we covered MySQL, an open source implementation of SQL that is used widely in companies like LinkedIn, Facebook, Google, Amazon and many more.

To install MySQL on your system, kindly follow the instructions provided in the installation document.

To connect to a MySQL database, you need a username, password, host ip address, port number and the database name. An internet connection is also required in case the server isn't your local system or on your local network.

Types of SQL Commands

The standard SQL commands are broadly classified into 3 types:

- DDL (Data Definition Language) – Deal with database schemas and description, defining how data resides in the database. Eg: ALTER, DROP, CREATE
- DML (Data Manipulation Language) – Deal with data manipulation in tables. Eg: SELECT, INSERT, UPDATE, DELETE.
- DCL (Data Control Language) – Deal with granting rights to user and revoking rights of control from users to the database. Eg: GRANT, REVOKE

Apart from these, there are a few other commands:

- TCL (Transaction Control Language) – Deals with saving and rolling back changes. Eg: ROLLBACK, COMMIT, SAVEPOINT.

SQL Commands are not case sensitive, so **SELECT** will be the same as **select.** The table and column names may or may not be case sensitive depending upon the system. Windows and Mac systems are case insensitive whereas Linux systems are usually case sensitive. To know more, click here.

Common SQL Commands - 1

This section includes commands used in the session on the world database.

SELECT – Chooses columns to retrieve by the query. Used for selecting fields.

Syntax: SELECT Column **FROM** Table;

Usage: SELECT fname **FROM** employee;

To select multiple columns, separate column names with a comma.

SELECT Column1, Column2, Column3 **FROM** Table;

To select all columns, use *.

SELECT * **FROM** Table;

WHERE – Subsets based on a condition. Used for selecting records.

Syntax: SELECT Column **FROM** TABLE **WHERE** Column2 = “some_value”;

Usage: SELECT fname **FROM** employee **WHERE** Dno **=** 5;

DISTINCT – Returns only distinct observations.

Syntax: **SELECT DISTINCT** Column **FROM** TABLE;

Usage: **SELECT DISTINCT** fname **FROM** employee;

AND / OR – Combines multiple conditions.

Syntax: SELECT Column **FROM** TABLE **WHERE** Column2 = “some_val” **AND / OR** Column3 = “some_val”;

Usage: SELECT * FROM works_on **WHERE** Pno = 1 **OR** Pno = 2;

SELECT * **FROM** projects **WHERE** Pno = 1 **AND** Dno = 5;

ORDER BY – Sorts the field.

Syntax: **SELECT** Column **FROM** TABLE **ORDER BY** Column1;

Usage: **SELECT** * **FROM** employee **ORDER BY** Dno;

To sort by descending order, use the keyword DESC.

Usage: **SELECT** * **FROM** employee **ORDER BY** Dno DESC;

LIKE – Used to subset based on a pattern, more precisely to say a partial match.

Syntax: SELECT Column **FROM** TABLE **WHERE** Column1 **LIKE** “PATTERN”;

Usage: SELECT * FROM department **WHERE** Dname **LIKE** “Ad%”;

Note that the “%” is used to match anything. If only records in which the Department name begins with Ad are needed, the pattern would be “Ad%” , meaning match Ad followed by anything.

IN – Used to check presence in a value or list of values.

Syntax: SELECT Column **FROM** TABLE **WHERE** Column1 **IN (**“VAL1”);

Usage: SELECT Dnumber **FROM** dept_location **WHERE** Dlocation IN ("Houston","Stafford");

BETWEEN – Returns all observations between a range.

Syntax: SELECT COLUMN **FROM** TABLE **WHERE** COLUMN1 **BETWEEN** A **AND** B;

Usage: SELECT fname, lname **FROM** employee **WHERE** salary **BETWEEN** 25000 **AND** 50000**;**

LIMIT – Sets a limit to the number of results retrieved.

Syntax: SELECT COLUMN **FROM** TABLE **LIMIT** n; (where n -> number of records to retrieve)

Usage: SELECT * **FROM** employee **LIMIT** 5**;**

JOIN – Joins two tables. Could be an inner, left or right join.

Syntax: SELECT COLUMNS

FROM TABLE1

INNER JOIN / LEFT JOIN / RIGHT JOIN TABLE2

ON TABLE1.COLUMN = TABLE2.COLUMN;

Usage: SELECT dname, avg(salary)

FROM employee

INNER JOIN department **ON** dno=dnumber

WHERE salary >= 30000

GROUP BY dname

ORDER BY dname;

OUTER JOIN Implementation:

SELECT e.ssn, e.fname, d.dependent_name **FROM** employee e

LEFT JOIN dependent d **ON** e.ssn = d.essn;

SELECT e.ssn, e.fname, d.dependent_name **FROM** department d

RIGHT JOIN employee e **ON** e.ssn = d.essn;

AGGREGATE FUNCTIONS – COUNT, MAX, MIN, AVG

USAGE: SELECT COUNT(*) FROM employee **WHERE** Dno = 5;

SELECT AVG(salary**) FROM** employee;

SELECT SUM(salary**) FROM** employee;

NESTED QUERIES: Query within a query.

Usage: SELECT fname, salary

FROM employee

WHERE salary >= (**SELECT** avg(salary) **FROM** employee);