

**NAGARJUNA COLLEGE OF ENGINEERING AND TECHNOLOGY**

**Department of CSE (Data Science)**

**FOUNDATION OF DATA SCIENCE USING SQL LAB MANUAL**

**22CDI32**

**PREPARED BY**

NIKHILA M R

ASSISTANT PROFESSOR

Department of CSE (Data Science)

**Create company database with 5 tables and insert values to the tables.**

* **CREATE**

The CREATE TABLE statement is used to create tables in SQL.

**Syntax:**

CREATE TABLE table\_name (column1 data\_type(size) constraint\_name,

column2 data\_type(size) constraint\_name,

column3 data\_type(size) constraint\_name,........);

* Create table employee(fname varchar(20), mname varchar(20), lname varchar(20), ssn varchar(20) primary key, bdate date, Address varchar(20),super\_ssn varchar(20), Dno integer);
* Create table department(dname varchar(20), dnumber integer primary key, mgrssn varchar(20),mgr\_start\_date date);
* Create table department\_location(dno integer, dlocation varchar(20));
* Create table project(pname varchar(20), pnumber integer primary key, Plocation varchar(20), Dno integer);
* Create table works\_on(Essn varchar(20), pnumber integer, hours integer);
* **INSERT**

The INSERT INTO statement of SQL is used to insert a new row in a table.

Syntax:

INSERT INTO table\_name VALUES (value1, value2, value3,…);

Example:

* Insert into employee values(‘anu’,’m’,’rao’,’88756’,’ 2-jan-1999’,’pune’,’87693’,’2’);
* Insert into employee values(‘gagan’,’A’,’dola’,’87693’,’ 3-feb-1999’,’delhi’,’88756’,’2’);

Employee:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Fname | Mname | Lname | Ssn | Bdate | Address | Super\_ssn | Dno |
| Anu | M | Rao | 88756 | 2-jan-1999 | Pune | 87639 | 2 |
| Gagan | A | Dola | 87693 | 3-feb-2000 | Delhi | 88756 | 2 |
| Sai | C | elkal | 76543 | 1-march-1996 | Dubai | 21016 | 3 |
| Pruthvi | B | Rao | 21016 | 5-dec-1990 | USA | 88756 | 4 |
| Vishwa | S | murthi | 78910 | 7-12-2001 | karnataka | 76543 | 3 |

Department:

|  |  |  |  |
| --- | --- | --- | --- |
| Dname | Dnumber | Mgrssn | Mgr\_ssn\_date |
| DS | 1 | 87693 | 10-may-1992 |
| AIML | 3 | 76543 | 2-jan-1999 |
| IS | 4 | 21016 | 3-feb-2000 |
| CS | 2 | 8875 | 1-march-1996 |
| EC | 5 | 78910 | 2-april-1999 |

Department\_location:

|  |  |
| --- | --- |
| Dnumber | Dlocation |
| 1 | Bangalore |
| 2 | Pune |
| 3 | Hyderabad |
| 4 | Lucknow |
| 5 | Mumbai |

Project:

|  |  |  |  |
| --- | --- | --- | --- |
| Pname | Pnumber | Plocation | Dno |
| Construction | 360 | karnataka | 2 |
| Unemployment | 143 | Kerala | 1 |
| Poverty | 162 | Himachal | 4 |
| Data analytic | 136 | Delhi | 3 |
| Electronics | 420 | madyapradesh | 5 |

Works\_on:

|  |  |  |
| --- | --- | --- |
| Essn | Pno | Hours |
| 88756 | 162 | 2 |
| 87693 | 143 | 10 |
| 21016 | 360 | 5 |
| 76549 | 136 | 3 |
| 78910 | 420 | 9 |

Implementation of DML(Data Manipulation Language) commands of SQL with suitable examples.

* Where clause
* Update
* Delete
* **Select command and Where clause:**

Select command is used to retrieve records from table

* The WHERE clause is used to filter records.
* It is used to extract only those records that fulfill a specified condition.

## Syntax

## SELECT column1*,* column2,... FROM table\_name WHERE condition*;*

## Example:

1. SELECT \* FROM Employee  
   WHERE ssn= 88756;
2. SELECT pname,pnumber FROM Project  
   WHERE pnumber > 162   ;

* **UPDATE:**

The UPDATE statement is used to modify the existing records in a table.

### Syntax:

UPDATE table\_name  
SET column1 *=* value1*,* column2 *=* value2*, ...*  
WHERE condition;

Example:

1. UPDATE Department  
   SET dname= 'research'  
   WHERE dnumber = 4;
2. UPDATE employee  
   SET fname='Juan'  
   WHERE lname=' Rao';

* **DELETE:**

The DELETE statement is used to delete existing records(rows) in a table.

**Syntax**

DELETE FROM table\_name WHERE condition*;*

Example:

1. DELETE FROM employee WHERE fname='Anu';
2. DELETE FROM works\_on;

Implementation of DDL(Data Definition Language) commands of SQL with suitable examples.

* Alter
* Drop
* **Alter Command:**

The ALTER TABLE statement is used to add, delete, or modify columns in an existing table.

### ALTER TABLE ADD Column Statement

### Syntax:

ALTER TABLE table\_name ADD Columnname\_1  datatype;

**Example:**

ALTER TABLE Employee ADD Email varchar(20);

**ALTER TABLE DROP Column Statement**

DROP COLUMN is used to drop columns in a table.

**Syntax**:

ALTER TABLE table\_name DROP COLUMN column\_name;

**Example:**

ALTER TABLE Employee DROP COLUMN Email;

* **DROP**
* The DROP TABLE statement is used to drop an existing table in a database.

### Syntax:

* DROP TABLE table\_name;

Example:

1. DROP TABLE Employee;
2. DROP TABLE project;

Implementation of Aggregate Functions with suitable examples.

* The COUNT() function returns the number of rows that matches a specified criterion.
* The AVG() function returns the average value of a numeric column.
* The SUM() function returns the total sum of a numeric column.
* The MAX() function returns the maximum number in the column.
* The MIN() function returns the maximum number in the column.

Syntax:

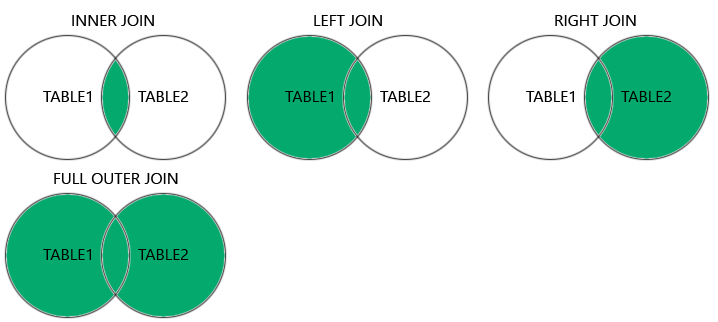
SELECT AVG(column\_name),COUNT(column\_name),MIN(column\_name)  
FROM table\_nameWHERE condition;

Example:

1. SELECT COUNT(\*)  
    FROM employee WHERE dno=2;
2. SELECT COUNT(salary),COUNT(salary),AVG(salary),MAX(salary)  
    FROM employee;

Implementation of different types of joins.

Join command combines rows from two or more tables based on related column.



* **Inner join:**

The INNER JOIN keyword selects records that have matching values in both tables.

**Syntax**:

Select column\_name(s)

From table1 INNER JOIN table2

ON table1.column\_name = table2.column\_name;

**Example:**

1. Select \* from department INNER JOIN dept\_location ON dno = dnumber;
2. Select \* from employee INNER JOIN department ON dno=dnumber;

* **Left join:**

The left join operation returns all record from left table and matching records from the right table. On a matching element not found in right table, NULL is represented in that case.

**Syntax:**

Select column\_name(s)

From table1 LEFT JOIN table2

ON table1.column\_name = table2.column\_name;

**Example:**

SELECT \*

FROM employee LEFT JOIN dapartment

ON Dno=Dnumber;

* **Right join:**

The left join operation returns all record from left table and matching records from the right table. On a matching element not found in right table, NULL is represented in that case.

**Syntax:**

Select column\_name(s)

From table1 RIGHT JOIN table2

ON table1.column\_name = table2.column\_name;

**Example:**

SELECT \*

FROM employee RIGHT JOIN dapartment

ON Dno=Dnumber;

* **Full Outer Join :**

The SQL FULL OUTER JOIN statement joins two tables based on a common column. It selects records that have matching values in these columns and the remaining rows from both of the tables.

**Syntax:**

Select column\_name(s)

From table1 FULL OUTER JOIN table2

ON table1.column\_name = table2.column\_name;

**Example:**

SELECT \*

FROM employee FULL OUTER JOIN dapartment

ON Dno=Dnumber;

**Note**: Refer foundation of datascience using SQL notes for examples.

**Study and implementation of group by ,order by commands.**

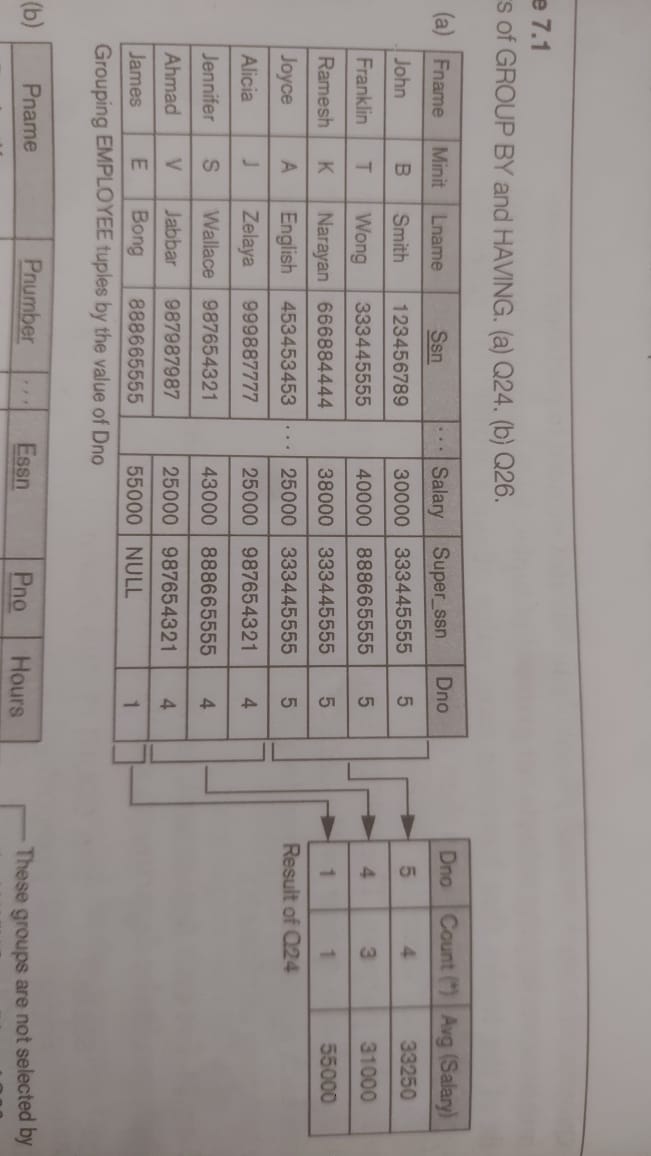
* **Group by:**
* The GROUP BY command is used to group the result set
* **Note**: The GROUP BY clause specifies the grouping attributes,which should also appear in select clause.

Q: For each department, retrieve the department number, the number of employees in the department and their average salary.

1. Select dno,count(\*),avg(salary)

From employee

Group by dno;



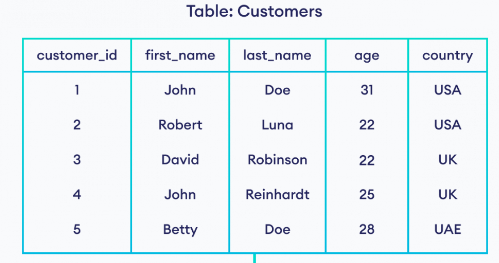
* HAVING condition compares the column to certain conditions that require filtering

1. Select count(customer\_id,country)

From customers

Group by country

Having count(customer\_id)>1;



Output:



* **Order by:**

The ORDER BY keyword is used to sort the result-set in ascending or descending order.

1. Select fname,lname,ssn

From employee

Order by fname asc;

1. Select fname,lname,ssn

from employee

Order by fname desc;

**Study and implementation of commit,savepoint,rollback.**

* **COMMIT command**

COMMIT command is used to permanently save any transaction into the database.

COMMIT;

* SAVEPOINT command

SAVEPOINT command is used to temporarily save a transaction so that you can rollback to that point whenever required.

Syntax: SAVEPOINT savepoint\_name;

Example:

UPDATE employee SET name = 'Abhijit' WHERE ssn=\_\_\_\_\_ ;

SAVEPOINT A;

* **ROLLBACK command**

This command restores the database to last commited state. It is also used with SAVEPOINT command to jump to a savepoint in an ongoing transaction.

UPDATE department SET dname = 'res' WHERE dnumber=2;

SAVEPOINT A;

UPDATE department SET dname = 'aaa' where dnumber=2;

SAVEPOINT B;

UPDATE department SET dname = 'bbb' where dnumber=3;

SAVEPOINT C;

ROLLBACK TO A;

SELECT \* FROM department;

ROLLBACK command to roll back the state of data to the **savepoint B**.