

CHAPTER 5

Module IV

Structured Query Language

Syllabus

Overview of SQL, Data Definition Commands, Data Manipulation commands, Data Control commands, Integrity constraints in SQL. Grant & Revoke Self-learning Topics: Physical design of database for the relational model designed in module III and fire various queries.

5.1 Overview of SQL

Q. Explain role SQL with example.

- **SQL (Structured Query Language)** is a computer language aimed to store, manipulate, and retrieve data stored in relational databases.
- It was developed by IBM Research in the mid 70's and standardized by ANSI in 1998.
- The first commercial relational database was released by Relational Software (later called as Oracle).
- SQL is a keyword based language and each statement begins with a unique keyword.
- SQL syntax is not case sensitive.

5.1.1 Role of SQL

1. SQL is an interactive query language which can be used to retrieve data from database.
2. SQL is a database programming language which can be used along with programming language to access data from database.
3. SQL is a database administration language which can be used to monitor and control data access by various users.
4. SQL can be used as an Internet data access language.

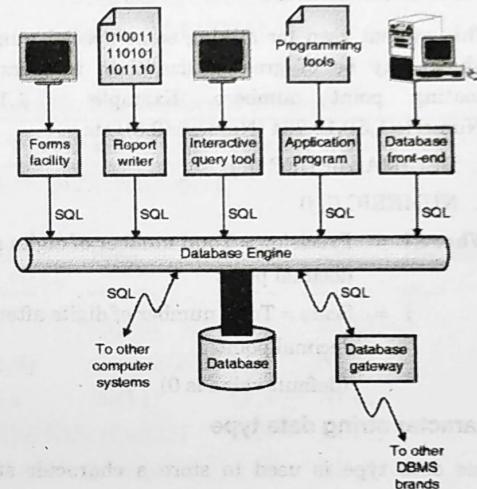


Fig. 5.1.1 : Role of SQL in DBMS

5.2 SQL Data Types

Q. Explain various SQL data types with example.

- The basic data types available with SQL standard are as enlisted below, all data types may not be supported by SQL server or Oracle.
- Data types include numeric, character string, bit string, Boolean and date time.

1. Numeric data types

This datatype is used to store a number values that can be decimal or floating point values.

a. Integer number of various size

- These types of system are used to store natural numbers which are not having any decimal values.
- **Example :** 111, 23 etc.

- Generally not supported by SQL server but supported by Oracle.

5.3 Data Definition Language (DDL) - Physical Design of The Relational Model (Self Learning Topic)

Q. Explain DDL commands with example.

MU - May 17, 10 Marks

Q. Write short note on DDL command.

MU - Dec/19, 5 Marks

- To create database schema and database objects like table **Data Definition Language (DDL)** can be used.
- DDL statements are used to build and modify the structure of your tables and other objects in the database.
- The set of DDL commands are as below,
 - CREATE Statement** : To create Database objects
 - ALTER Statement** : To modify structure of database objects
 - DROP Statement** : To remove database objects
 - RENAME Statement** : To Rename Database objects
 - TRUNCATE Statement** : To empty the database table
- When you execute a DDL statement, it takes effect immediately, as it is **Autocommitted** into database. Hence no rollback operation (Undo) can be performed with these set of commands.
- Database objects are any data structure created in database.
- E.g. Table, View, Sequence etc.

5.4 CREATE Statement / CREATE Table

Q. Explain CREATE command with example.

- CREATE statement is used to create new database objects like table, index and others.
- CREATE TABLE is the command in database system is used to create a new table with unique name or identifier.
- This statement used to create database object.

Syntax

CREATE TABLE <Table_Name>

```
( Column_1 datatype,  
Column_2 datatype,  
....  
Column_n datatype  
);
```

Example

SQL> CREATE TABLE Employee

```
( Eid INT,  
Name VARCHAR (20),  
Age INT,  
Address CHAR (25),  
Salary DECIMAL (18, 2)  
);
```

Query OK, 0 rows affected (0.01 sec)

- To view the structure of newly created table.

SQL> DESC Employee;

Field	Type	Null	Key	Default	Extra
EID	int(10)	YES	NULL		
NAME	varchar(20)	YES	NULL		
AGE	int(11)	YES	NULL		
ADDRESS	char(25)	YES	NULL		
SALARY	decimal(18,2)	YES	NULL		

5 rows in set (0.00 sec)

5.5 Create Table with Constraints - Integrity Constraints in SQL (Self Learning Topics)

5.5.1 Domain Integrity Constraint

Domain constraints are used to test the values inserted into the table is correct or not.

a. Required Data Constraint / Not Null Constraint

- Some attributes (columns) in a database are not allowed to contain NULL value. NULL values are values which are unknown, unassigned or missing attribute values.
- Example :** In the student database, every student must have an associated student name. Student_name should not be NULL.



```
SQL> CREATE TABLE Student
      ( Eid   INT NOT NULL);
Query OK, 0 rows affected (0.01 sec)
```

b. Check Constraint

- Use of **check constraint** is to ensure that attribute value satisfies specific user defined condition.
- Example :** Table with customer entity having name, cid and gender which can be M or F.
- Hence, attribute gender can take only two values either 'M' or 'F'.

```
SQL> CREATE TABLE customer
      ( Name  CHAR (25) NOT NULL,
        Gender CHAR (1),
        CHECK (Gender IN ('M', 'F'))
      );
Query OK, 0 rows affected (0.01 sec)
```

c. Default Keyword

- Default keyword is used to add some value if no attribute value added for tuple.
- Example :** Table with customer entity having name, cid and gender in which cid is primary key. If name is not added for customer that will be taken as 'Unknown'.

```
SQL> Create table customer
      ( Name  char (25) DEFAULT 'UNKNOWN';
Query OK, 0 rows affected (0.01 sec)
```

5.5.2 Entity Integrity Constraint

Entity constraints are used to test the values inserted into the database are correct or not with respect to other tuples in same table.

a. Primary Key Constraint

- Primary key attribute is same as unique key constraint with Not NULL constraints.
- Primary key attribute values needs to be unique as well as null values are not allowed in primary key attributes.
- The main difference in unique constraint and primary key constraint is that one null value is allowed in unique constraint which can be treated as unique value while nulls are not allowed in primary key constraint.

- Example :** Table with customer entity having name, cid and gender in which cid is primary key.

```
SQL> CREATE TABLE customer
      ( Name  CHAR (25),
        Cid  CHAR (10) PRIMARY KEY
      );
Query OK, 0 rows affected (0.01 sec)
```

b. Unique Constraint

- In case of unique constraint no two tuples can have equal value for same attributes.
- This constraint says that attributes forms candidates key, which allows one Null value which is unique by itself.
- This UNIQUE constraint can be applicable to user defined domain declaration also.

Example

```
SQL> CREATE TABLE customer
      ( Name  CHAR (25),
        Cid  CHAR (10),
        Email  CHAR (50) UNIQUE
      );
Query OK, 0 rows affected (0.01 sec)
```

5.5.3 Referential Integrity Constraint in SQL

- A value appearing in a one relation (table) for a given set of attributes also appears for another set of attributes in another relation (table).
- This is called referential integrity.
- The referential integrity constraint is defined between two tables to maintain the consistency among tuples in the two relations.

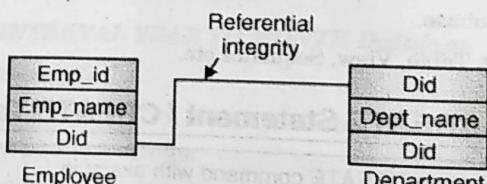


Fig. 5.5.1

Example

```
SQL> Create table Department
      ( Did INT,
        Dept_name VARCHAR (100) NOT NULL,
```

```

PRIMARY KEY (did)
);

Query OK, 0 rows affected (0.01 sec)

SQL> Create table Emp
(
    Emp_id INT,
    Emp_name VARCHAR (100) NOT NULL,
    Did     INT REFERENCES department (did)
);
Query OK, 0 rows affected (0.01 sec)

```

5.6 Alter Table

Q. Explain ALTER command with example.

- Once database object is created in database, we may require to the ALTER command is used to update structure of database object.
- The ALTER TABLE statement can be used to add, delete, or modify columns in an existing table.
- The ALTER TABLE command can also be used to add and drop various constraints on an existing table.

Syntax

ALTER TABLE <Table_Name>

ADD Column_1 datatype;

ALTER TABLE <Table_Name>

Modify Column_1 New_datatype;

ALTER TABLE <Table_Name>

DROP Column_1;

Example

```
SQL> ALTER TABLE Employee
      ADD Address VARCHAR (100);
```

Query OK, 0 rows affected (0.01 sec)

To view the changed structure of table

```
SQL> DESCRIBE TABLE Employee;
```

Field	Type	Null	Key	Default	Extra
EID	int(10)	YES	NULL		
NAME	varchar(20)	YES	NULL		
AGE	int(11)	YES	NULL		
ADDRESS	varchar(100)	YES	NULL		
SALARY	decimal(18,2)	YES	NULL		

5 rows in set (0.00 sec)

5.7 Rename Table

Q. Explain RENAME command with example.

- It is possible to change name of table with or without data in it using simple RENAME command.
- We can rename any table object at any point of time.

Syntax

RENAME TABLE <Table_Name> To <New_Table_Name>;

Example

```
SQL> RENAME TABLE Employee To EMP;
```

5.8 Truncate Table

Q. Explain TRUNCATE command with example.

- The SQL TRUNCATE TABLE command is used to delete all data from an existing table.
- It is possible to do same action with DROP TABLE command but it would remove complete table structure from the database.
- A DELETE command will also remove all data from table but with DELETE data deletion can be rolled back and truncate acts as permanent data deletion with no roll back possible.
- If any delete triggers are defined on the table, then the triggers are not fired on truncate table.
- Truncate will de-allocates memory space. So that the free space can be used by other tables unlike DELETE command.

Syntax

TRUNCATE TABLE <Table_Name>;

Example

```
SQL> TRUNCATE TABLE EMP;
```

5.9 Drop Command / DROP Table

Q. Explain DROP command with example.

- Drop command can be used to remove database any objects from user database.



- The SQL DROP TABLE statement is used to remove a table definition and all related data like indexes, triggers, constraints and permission specifications for that table.
- The developer must be careful while running this command because once a table is dropped then all the information available in that table will also be lost forever and no roll back can be done.

Syntax

```
DROP TABLE <Table_Name>;
```

Example

- If we want to permanently remove the Employee table that we created, we'd use the following command

```
SQL> DROP TABLE Employee;
Query OK, 0 rows affected (0.01 sec)
```

- If we want to permanently remove the Employee table that we created, we'd use the following command

```
SQL> DESC TABLE Employee;
ERROR 1146 (42S02) : Table 'TEST.Employee' doesn't exist
```

5.10 Data Manipulation Language (DML)

Q. Explain DML commands with syntax.

MU - May 17, 10 Marks

- Data Manipulation Language (DML)** statements are used for manipulating or managing data in database.
- DML commands are not auto-committed like DDL statements.
- It means changes done by DML command can be rolled back. Or in other words the DML statements do not implicitly commit the current transaction.
- DML is set of commands used to,
 - Insert data into table
 - Delete data from table
 - Update data of table

5.10.1 INSERT Statement

- Insert statement used to add records to the existing table.
- To insert data into a table, SQL INSERT INTO command can be used.

- To insert few values in table as per columns names we can use generic syntax as below,

```
INSERT INTO <Table_Name> (Column1, . . . , ColumnN)
```

```
VALUES (column1 . . . , columnN);
```

- If all values for all the columns of the table are to be added then also no need to specify the column names in the SQL query.
- But, we need to make sure the order of the values is in the same order as the columns in the table.

```
INSERT INTO <Table_Name>
```

```
VALUES (column1 . . . , columnN);
```

Example

```
SQL> INSERT INTO Employee VALUES (1001, 'Mahesh');
SQL> INSERT INTO Employee VALUES (NULL, 'Jayendra');
```

```
SQL> INSERT INTO Employee (Name, Eid) VALUES ('Sachin', 1002);
```

```
SQL> INSERT INTO Employee (Name, Eid) VALUES ('Suhas', NULL);
```

- The data added in table can be displayed as follows,

```
SQL> SELECT *  
      FROM Employee;
```

EID	NAME
---	-----
1001	Ramesh
1002	Khilan
NULL	Kaushik
NULL	Chaital

5.10.2 DELETE Statement

- Delete statement is used to delete some or all records from the existing table.
- To delete data into a table, SQL DELETE command can be used.
- To delete all rows in table we can use generic syntax as below,

Syntax

```
DELETE
FROM <Table_Name>;
```

To delete selected rows from table we can specify the WHERE condition.

```
DELETE
FROM <Table_Name>
WHERE <Condition>;
```

Example

```
DELETE
FROM Employee
WHERE Eid IS NULL;
```

To check rows deleted by above delete query is as given.

```
SELECT *
FROM Employee;
SQL> SELECT *
      FROM Employee;
```

EID	NAME
1001	Ramesh
1002	Khilan

5.10.3 UPDATE Statement

- The UPDATE statement is used to modify the existing data present in a table.
- To update data in a table, SQL UPDATE command can be used.
- To update all rows in table we can use generic syntax as below,

```
UPDATE <Table_Name>
SET column1 = new_value;
```

- To update selected rows from table we can specify the WHERE condition in Update statement.

```
UPDATE <Table_Name>
SET column1 = new_value
WHERE condition;
```

Example

```
SQL> UPDATE Employee
      SET Eid = 1002
      WHERE name = 'Suhas';
```

- To check rows updated in table using select query,

```
SQL> SELECT *
      FROM Employee;
```

```
SQL> SELECT *
      FROM Employee;
```

EID	NAME
1001	Ramesh
1002	Suhas

5.11 Data Control Language (DCL)

- Q. Write a short note on : DCL commands.

MU - Dec. 17, 5 Marks

- Q. Write a note on DCL.

- Data Control Language (DCL)** is used to control various user actions (or privileges) in Database.
- To perform any operation in the database user needs **privileges** like creating tables, sequences or views.
- DCL is set of commands used to,
 - Grant** : Gives some privilege to user for performing task on database.
 - Revoke** : Take back permissions given from user.
- Privileges can be of many types,
 - System Privileges** : creating a table is types of system privilege.
 - Object Privileges** : To execute query on tables object privilege can be used.
 - Ownership Privileges** : To execute query on tables created by same user.

5.12 Privileges

- Q. Explain all types of privileges.

- Q. Enlist the privileges with suitable example

- Q. Write a short note on : 1. Ownership Privileges
2. Object Privileges 3. System Privileges



1. Introduction

- The set of actions that a user can perform on a database object are called the privileges.
- Privilege is right to execute particular SQL statement on database.
- The high level user (Like DBA) has power to grant access to database and its object.

2. System privileges

- System privileges are rights and restriction that are implemented on databases to control which users can access how much data in the database.
- User requires system privileges to gain access to database.
- System privileges are generally provided by DBA.
- Few system privileges are as below,

System privileges	Authorized to
CREATE USER	Create number of users in DBMS
DROP USER	Drop any other users in DBMS
CREATE ANY TABLE	Create table object in any schema.
SELECT ANY TABLE	Query table object or view in any schema.
DROP ANY TABLE	Drop table object in any schema.

3. Object privileges

- Object privileges are rights and restrictions to change contents of database objects.
- User requires object privileges to manipulate the content of object within database.
- Once we have created object in a database, after some time there may be few changes needs to be introduced in object.
- Not all database users are allowed to make such changes in database; hence administrator should have control over all objects modification.
- The user which has GRANT ANY PRIVILEGE system privilege granted to him then he can act like administrator to control database modifications.
- Different objects has different privileges assigned for him,
- Few object privileges are as below,

Object privileges	Authorized to
SELECT	Select rows from table or view
INSERT	Add new rows to table or view
DELETE	Remove some rows from table or view
UPDATE	Modify content of rows from table or view
EXECUTE	To run procedure
REFERENCES	To reference a particular table using foreign key and check constraint

4. Ownership privileges

- Whenever you create a database object (like table or view) with the CREATE statement, you will become its owner and get full privileges for the table. (Like SELECT, INSERT, DELETE, UPDATE, and all other privileges)
- All other users are having no privileges on the newly created database object.
- You as owner of database object can explicitly give grant privileges to any other user by using the GRANT statement.
- Whenever you are creating a view with the CREATE VIEW statement, you become the owner of that view, but you do not necessarily receive all privileges as you require the SELECT privilege on each of base tables on which view is defined.

5.13 Granting Privileges

Q. Write syntax for GRANT privileges.

1. Introduction

- A system privilege is the right to perform a particular action, or to perform an action on any schema objects of a particular type.
- An authorized user may pass on this authorization to other users. This process is called as granting of privileges
- Generally GRANT statement is used by owner of table or view to give other users access permissions.

- In SQL user accounts must present in system before we can grant privileges to him.

2. Syntax

```
Grant <ALL | privilege list>
ON <relation name or view name>
TO <user | role list | PUBLIC >
[WITH GRANT OPTION]
```

Privilege list	Meaning
ALTER	Table and views
CREATE	Table and views
DROP	Table and views
DELETE	Tables and views
INSERT	Tables and views
SELECT	Tables and views
UPDATE	Tables and views
ALL	Tables and views

WITH GRANT OPTION

Is used to allow user to grant privileges (which are granted to him) to other users.

3. Example

- Consider an example for granting update authorization to the *Emp_Salary* relation of the company database. Assume that, initially that the DBA grants update authorization on *Emp_Salary* to other users U1, U2 and U3, who may in turn pass on this authorization to other users. This passing of authorization from one user to other users is called **authorization graph**.
- The following **grant** statement grants user U1, U2 and U3 the **select** privilege on *Emp_Salary* relation

```
GRANT SELECT, INSERT
ON mydb.*
TO 'mahesh'@'somehost';
```

- Following **grant** statement gives all users all authorization on the amount attributes of the *Emp_Salary* relation using public keyword;

```
GRANT ALL
ON *.*
TO 'mahesh'@'somehost';
```

a. Database privileges

- SQL permits a user to declare foreign keys while creating relations.
- Example** Allow user U1 to create relation that references key 'Eid' of Emp_Salary relation.

```
GRANT ALL
```

```
ON mydb.* TO 'mahesh'@'somehost';
```

b. Table privileges

This privilege authorizes a user to execute a function or procedure.

```
GRANT ALL
```

```
ON mydb.mytbl
```

```
TO 'mahesh'@'somehost';
```

```
GRANT SELECT, INSERT
```

```
ON mydb.mytbl
```

```
TO 'mahesh'@'somehost';
```

c. Column privileges

This privilege authorizes a user to execute a function or procedure.

```
GRANT SELECT (col1), INSERT (col1, col2)
```

```
ON mydb.mytbl
```

```
TO 'mahesh'@'somehost';
```

5.14 Revoking of Privileges

Q. Write a short note on Revoking of privileges.

1. Introduction

- We can reject the privileges given to particular user with help of revoke statement.
- To revoke an authorization, we use the **REVOKE** statement.

2. Syntax

```
REVOKE <ALL | privilege list>
```

```
ON <relation name or view name>
```

```
FROM <user | role list | PUBLIC>
```

```
[RESTRICT/ CASCADE]
```

- CASCADE** : will revoke all privileges along with all dependent grant privileges
- RESTRICT** : This will not revoke all related grants only removes that GRANT only.



3. Examples

- The revocation of privileges from user or role may cause other user or roles also have to leave that privilege.
- This behaviour is called cascading of the revoke.
 - To remove select privilege from users U1, U2 and U3.

REVOKE SELECT

```
ON      mydb.mytbl
FROM   'mahesh'@'somehost';
```

- To remove update rights on amount column of Emp_Salary from U1, U2 and U3.

REVOKE UPDATE (amount)

```
ON      Emp_Salary
FROM   'mahesh'@'somehost';
```

- To remove reference right on amount column from user U1.

REVOKE REFERENCES (amount)

```
ON      Emp_Salary
FROM   'mahesh'@'somehost';
```

The revoke statements may alternatively specify restrict if we don't want cascade behavior.

REVOKE SELECT

```
ON      Emp_Salary
FROM   'mahesh'@'somehost'
RESTRICT;
```

5.15 Solved Designing Problem

Example 5.15.1 : For the given database, write SQL queries.

Employee (Eid, Name, Street, City)

Works (Eid, Cid, salary)

Manager (Eid, Manager_Name)

Company(Cid, Company_name, city)

Solution :

- Modify the database so that 'Jack' now lives in 'Newyork'.

```
MySQL> UPDATE Employee
      SET     City = 'Newyork'
      WHERE   Name = 'Jack';
```

- Give all employees of 'ANZ corporation' a 10% raise in salary.

MySQL> UPDATE Works

```
SET     Salary = (salary + (0.1 * salary))
```

```
WHERE   CID IN ( SELECT Cid
```

```
FROM    Company
```

```
WHERE Company_name = 'ANZ corporation');
```

Example 5.15.2 : For the following given database ? Write SQL queries

person (driver_id #, name, address)

car (license, model, year)

accident (report_no, date, location)

owns (driver_id #, license)

participated (driver_id, car, report_number,

damage_amount)

Solution :

- Update the damage amount for car with licence number "Mum2022" in the accident with report number "AR2197" to Rs. 5000.

MySQL> UPDATE Participated

```
SET     Damage_amount = 500
```

```
WHERE   Report_number LIKE 'AR2197'
```

```
AND Car = 'Mum2022';
```

Example 5.15.3 : For given database, write SQL queries.

Employee (EID, Name, Street, City)

Works (EID, CID, Salary)

Manager (EID, Manager_Name)

Company (CID, Company_name, City)

Solution :

- Modify the database so that 'TRATHAM' now lives in USA

MySQL> UPDATE Employee

```
SET     City = 'USA'
```

```
WHERE   Name = 'TRATHAM';
```

- Give all employees of 'SHARAYU Steel' a 10% raise in salary.

MySQL> UPDATE works

```
SET     salary = (salary + (0.1 * salary))
```

```
WHERE   cid IN ( SELECT cid
```

```
FROM    company
```

```
WHERE company_name = 'SHARAYU Steel');
```

Example 5.15.4 : Consider insurance database given below and answer the following queries in SQL.

Person (driver_id, name, address)

Car (license, model, year)

Accident (report_no, adate, location)

Owns (driver_id, license)

Participated (driver_id, license, report_no, damage_amount)

MU - Dec. 14, 10 Marks

Solution :

1. Add new accident to database.

```
MySQL> INSERT INTO Accident (report_no, adate, location)
VALUES ('111', '01/01/2014', 'Pune');
```

2. Delete 'Santro' belonging to 'John Smith'.

```
MySQL> DELETE
      FROM CAR
      WHERE Model = 'SANTRO'
      AND
      License IN ( SELECT license
      FROM Owns
      WHERE Driver_id IN (SELECT driver_id
      FROM person
      WHERE name = 'John
      Smith')
      );
```

Example 5.15.5 : Consider the following employee database.

Employee (empname, street, city, date_of_joining)

Works (empname, company_name, salary)

Company (company_name, city)

Manages (empname, manager_name).

Write SQL queries for the following statements :

1. Modify the database so that 'John' now lives in 'Mumbai'.

2. Give all employees of ABC Corporation' a 10% raise

MU - Dec. 13, 5 Marks

Solution :

- (i) Modify the database so that 'John' now lives in 'Mumbai'.

```
MySQL> UPDATE Employee
      SET City= 'Mumbai'
      WHERE Empname= 'JOHN';
```

2. Give all employees of ABC Corporation' a 10% raise.

```
MySQL> UPDATE Works
      SET Salary=0.1*salary
      WHERE Company_name= 'ABC Corporation';
```

Example 5.15.6 : Employees (Empid, Fname, Lname, Email,

Phoneno, Hiredate, Jobid, Salary, Mid, Did)

Departments (Did, Dname, Managerid, Locationid)

Locations (Locationid, Streetadd, Postalcode, City)

Write the SQL queries for the following.

1. List the employees have a manager who works for a department based in the U.S.
2. Write a query to display the details of all employees in the Finance department.
3. Give 10% hike to all the employees working in Did 20.
4. Write a query to display all the information of the employees whose salary is within the range 1000 and 3000.
5. Display the information of all the employees whose first name starts with 'R' in descending order of their salary.

Solution :

1. List the employees have a manager who works for a department based in the U.S.

```
SELECT *
  FROM Employees e
  INNER JOIN Departments d ON e.did = d.did
  INNER JOIN Locations l ON
  l.locationid=d.locationid
  WHERE City ='US';
```

2. Write a query to display the details of all employees in the Finance department.

```
SELECT *
  FROM Employees e
  INNER JOIN Departments d ON e.did = d.did
  WHERE Dname='Finance';
```

3. Give 10% hike to all the employees working in Did 20.

```
UPDATE employees
SET Salary = 1.1* Salary
WHERE did =20;
```

4. Write a query to display all the information of the employees whose salary is within the range 1000 and 3000.



```
SELECT *
FROM employees
WHERE Salary BETWEEN 1000 AND 3000;
```

5. Display the information of all the employees whose first name starts with 'R' in descending order of their salary.

```
SELECT *
FROM employees
WHERE Fname LIKE 'R%'
ORDER BY Salary DESC;
```

Review Questions

1. Explain various data definition statements in SQL.
2. Explain various data types used in SQL.
3. How to create table employee (Eid, Ename, age, DOB, address).
4. Explain Describe table command with example.
5. Explain various data definition statements in SQL.
6. Explain various data types used in SQL.
7. What is SQL ? Explain the following structures of SQL queries with appropriate example
 1. Select clause
 2. Where clause
 3. From clause
8. Write short note on : DDL and DML.
9. Write syntax for GRANT privileges.

10. Write a short note on Revoking of privileges.

5.16 University Questions and Answers

May 2017

1. Explain DML and DDL commands with syntax. (10 Marks)
2. Employees (Empid, Fname, Lname, Email, Phoneno, Hiredate, Jobid, Salary, Mid, Did)
Departments (Did, Dname, Managerid, Locationid)
Locations (Locationid, Streetadd, Postalcode, City)

Write the SQL queries for the following.

1. List the employees have a manager who works for a department based in the U.S.
2. Write a query to display the details of all employees in the Finance department.
3. Give 10% hike to all the employees working in Did 20.
4. Write a query to display all the information of the employees whose salary is within the range 1000 and 3000.
5. Display the information of all the employees whose first name starts with 'R' in descending order of their salary.

(10 Marks)

Dec. 2017

3. Write a short note on : DCL commands. (5 Marks)

Dec. 2019

4. Write short note on DDL command. (5 Marks)