SQL

SQL Basic queries:

Table name and columns
EMPLOYEE (empID, empName, salary, dept_id)

—Select Queries:
Select id,empId, empName, salary from EMPLOYEE;

*Select * from EMPLOYEE where empID = 101;*

*Select * from EMPLOYEE where salary> 10000;*

—Query to fetch first record from Employee table

*Select * from EMPLOYEE where Rownum = 1;*
Or
*Select * from EMPLOYEE where Rowid = select min(Rowid) from EMPLOYEE ;*

— Query to fetch last record from the table
*Select * from EMPLOYEE where Rowid = select max(rowid) from EMPLOYEE;*

—Query to display first 5 Records from Employee table
*Select * from Employee where Rownum <= 5;*

—Query to display last 5 Records from Employee table

*Select * from EMPLOYEE where rowid > (select count(*) from EMPLOYEE) - 5;*

—Query to find maximum salary of each department
SELECT dept_id, max(salary) from EMPLOYEE group by dept_id;

—Query to display Nth Record from Employee table?

*Select * from Employee where rowid =*
(Select count(*) from Employee)
-
((Select count(*) from Employee) - N)

—Query to find the second highest salary of the employee
SELECT emp_id, MAX(salary) AS salary
FROM EMPLOYEE WHERE salary < (SELECT MAX(salary) FROM EMPLOYEE);

— Query to find Nth highest Salary
Select top 1 from
(Select top N salary from Employee order by salary desc)
Order by salary asc;

—Query to to Display Odd rows in Employee table
*Select * from(Select rownum as rno,E.* from Employee E) where Mod(rno,2)=1;*

—Query to create table with same structure with data of Employee table
*Create table Employee1 as select * from Employee;*

—Query to fetch only common records between 2 tables.

*Select * from Employee;*
Intersect
*Select * from Employee1;*

—Query to get information of Employee where Employee is not assigned to the department

*Select * from Employee where Dept_no Not in(Select Department_no from Department);*

—Query to Select all records from Employee table whose name is 'Amit' and 'Pradnya'
*Select * from Employee where Name in('Amit','Pradnya');*

—Query to to find count of duplicate rows
Select rollno, count (rollno) from Student
Group by rollno
Having count (rollno)>1
Order by count (rollno) desc;

—Subqueries with the SELECT, UPDATE, DELETE Statements

*Select * from EMPLOYEE where id in (select id from EMPLOYEE where salary > 10000);*

*INSERT INTO EMPLOYEE_NEW SELECT * FROM EMPLOYEE WHERE ID IN (SELECT ID FROM EMPLOYEE) ;*

*UPDATE EMPLOYEE SET SALARY = SALARY * 0.25 WHERE ID IN (SELECT ID FROM EMPLOYEE_NEW WHERE ID >= 20);*

—Types of SQL Subqueries
Single Row Subquery. Returns zero or one row in results.
Multiple Row Subquery. Returns one or more rows in results.
Multiple Column Subqueries. Returns one or more columns.
Correlated Subqueries. ...
Nested Subqueries.

—Correlated SubQuery
A correlated subquery is evaluated once for each row processed by the parent statement. The parent statement can be a SELECT, UPDATE, or DELETE statement.
Example:1
Find all the employees who earn more than the average salary in their department.

```
SELECT last_name, salary, department_id
 FROM employees outer
 WHERE salary >
            (SELECT AVG(salary)
             FROM employees
             WHERE department_id =
                  outer.department_id);
```

Example:2
employee

employee_id	payment_type	amount_paid	payment_date

payment_history

employee_id	payment_type	amount_paid	payment_date

Obtain the names of employees who never received an award

```
SELECT last_name, first_name
FROM    employee e1
WHERE NOT EXISTS (SELECT ph.last_name
                        FROM payment_history ph
                        WHERE ph.employee_id = e1.employee_id
                        AND ph.payment_type = 'award')
```

Obtain the names of employees who received award payments.

```
SELECT    first_name, last_name
FROM      employee e1
          JOIN payment_history ph ON  ph.employee_id = e1.employee_id
```

WHERE ph.payment_type =award'

— JOINS
Inner Join:
The joining of two or more tables is based on common field between them.
Staff table

ID	Staff_NAME	Staff_AGE	STAFF_ADDRESS	Monthley_Package

Payment table

Payment_ID	DATE	Staff_ID	AMOUNT

SELECT Staff_ID, Staff_NAME, Staff_AGE, AMOUNT FROM STAFF s, PAYMENT p WHERE s.ID =p.STAFF_ID;

LEFT JOIN or Left outer join
The SQL left join returns all the values from the left table and it also includes matching values from right table, if there are no matching join value it returns NULL.

CUSTOMER TABLE:

ID	*NAME*	*AGE*	*SALARY*

ORDER TABLE:

O_ID	DATE	CUSTOMER_ID	AMOUNT

1. *SQL SELECT ID, NAME, AMOUNT,DATE*
2. *FROM CUSTOMER*
3. *LEFT JOIN ORDER*
4. *ON CUSTOMER.ID = ORDER.CUSTOMER_ID;*

RIGHT JOIN
The SQL right join returns all the values from the rows of right table. It also includes the matched values from left table but if there is no matching in both tables, it returns NULL

1. *SELECT ID,NAME,AMOUNT,DATE*
2. *FROM CUSTOMER*
3. *RIGHT JOIN ORDER*
4. *ON CUSTOMER.ID = ORDER.CUSTOMER_ID;*

FULL JOIN
The SQL full join is the result of combination of both left and right outer join and the join tables have all the records from both tables. It puts NULL on the place of matches not found.

SQL full outer join:

SQL full outer join is used to combine the result of both left and right outer join and returns all rows (don't care its matched or unmatched) from the both participating tables.

1. *SELECT **
2. *FROM table1*
3. *FULL OUTER JOIN table2*
4. *ON table1.column_name = table2.column_name;*

Cross Join
When each row of first table is combined with each row from the second table, known as Cartesian join or cross join.

1. *SELECT * FROM [TABLE1] CROSS JOIN [TABLE2]*
2. *OR*
3. *SELECT * FROM [TABLE1] , [TABLE2]*