**1. What are tables and Fields?**

A table is a set of data that are organized in a model with Columns and Rows. Columns can be categorized as vertical, and Rows are horizontal. A table has specified number of column called fields but can have any number of rows which is called record.

Example:.

Table: Employee.

Field: Emp ID, Emp Name, Date of Birth.

Data: 201456, David, 11/15/1960.

**2. What is a primary key?**

A primary key is a combination of fields which uniquely specify a row. This is a special kind of unique key, and it has implicit NOT NULL constraint. It means, Primary key values cannot be NULL.

**3. What is a unique key?**

A Unique key constraint uniquely identified each record in the database. This provides uniqueness for the column or set of columns.

A Primary key constraint has automatic unique constraint defined on it. But not, in the case of Unique Key.

There can be many unique constraint defined per table, but only one Primary key constraint defined per table.

 Multiple values allowed per table.

 Null values allowed.

**4. What is a foreign key?**

A foreign key is one table which can be related to the primary key of another table. Relationship needs to be created between two tables by referencing foreign key with the primary key of another table.

**5. What is a join?**

This is a keyword used to query data from more tables based on the relationship between the fields of the tables. Keys play a major role when JOINs are used.

**6. What are the types of join and explain each?**

There are various types of join which can be used to retrieve data and it depends on the relationship between tables.

* **Inner Join.**

Inner join return rows when there is at least one match of rows between the tables.

* **Right Join.**

Right join return rows which are common between the tables and all rows of Right hand side table. Simply, it returns all the rows from the right hand side table even though there are no matches in the left hand side table.

* **Left Join.**

Left join return rows which are common between the tables and all rows of Left hand side table. Simply, it returns all the rows from Left hand side table even though there are no matches in the Right hand side table.

* **Full Join.**

Full join return rows when there are matching rows in any one of the tables. This means, it returns all the rows from the left hand side table and all the rows from the right hand side table.

**7. What is the difference between DELETE and TRUNCATE statements?**

|  |  |
| --- | --- |
| **DELETE** | **TRUNCATE** |
| Delete command is used to delete a row in a table. | Truncate is used to delete all the rows from a table. |
| You can rollback data after using delete statement. | You cannot rollback data. |
| It is a DML command. | It is a DDL command. |
| It is slower than truncate statement. | It is faster. |

**8. What is the difference between DROP and TRUNCATE commands?**

DROP command removes a table and it cannot be rolled back from the database whereas TRUNCATE command removes all the rows from the table.

**Table - EmployeeDetails**

|  |  |  |  |
| --- | --- | --- | --- |
| **EmpId** | **FullName** | **ManagerId** | **DateOfJoining** |
| 121 | John Snow | 321 | 01/31/2014 |
| 321 | Walter White | 986 | 01/30/2015 |
| 421 | Kuldeep Rana | 876 | 27/11/2016 |

**Table - EmployeeSalary**

|  |  |  |
| --- | --- | --- |
| **EmpId** | **Project** | **Salary** |
| 121 | P1 | 8000 |
| 321 | P2 | 1000 |
| 421 | P1 | 12000 |

**Ques.1. Write a SQL query to fetch the count of employees working in project 'P1'.**

**SELECT** **COUNT**(\*) **FROM** EmployeeSalary **WHERE** Project = 'P1';

**Ques.2. Write a SQL query to fetch employee names having salary greater than or equal to 5000 and less than or equal 10000.**

**SELECT** FullName

**FROM** EmployeeDetails

**WHERE** EmpId **IN**

(**SELECT** EmpId **FROM** EmpolyeeSalary

**WHERE** Salary **BETWEEN** 5000 **AND** 10000);

**Ques.3. Write a SQL query to fetch project-wise count of employees sorted by project's count in descending order.**

**SELECT** Project, **count**(EmpId) EmpProjectCount

**FROM** EmployeeSalary

**GROUP** **BY** Project

**ORDER** **BY** EmpProjectCount **DESC**;

**Ques.4. Write a query to fetch only the first name(string before space) from the FullName column of EmployeeDetails table**

**mySQL- Using MID**

**SELECT** MID(FullName, 0, LOCATE(' ',FullName)) **FROM** EmployeeDetails;

**SQL Server-Using SUBSTRING**

**SELECT** SUBSTRING(FullName, 0, CHARINDEX(' ',FullName)) **FROM** EmployeeDetails;

**Also, we can use LEFT which returns the left part of a string till specified number of characters.**

**SELECT** **LEFT**(FullName, CHARINDEX(' ',FullName) - 1) **FROM** EmployeeDetails;

**Ques.5. Write a query to fetch employee names and salary records. Return employee details even if the salary record is not present for the employee.**

**SELECT** E.FullName, S.Salary

**FROM** EmployeeDetails E **LEFT** **JOIN** EmployeeSalary S

**ON** E.EmpId = S.EmpId;

**Ques.6. Write a SQL query to fetch all the Employees who are also managers from EmployeeDetails table.**

**SELECT DISTINCT** E.FullName

**FROM** EmpDetails E

**INNER** **JOIN** EmpDetails M

**ON** E.EmpID = M.ManagerID;

**Ques.7. Write a SQL query to fetch all employee records from EmployeeDetails table who have a salary record in EmployeeSalary table.**

**SELECT** \* **FROM** EmployeeDetails E

**WHERE** **EXISTS**

(**SELECT** \* **FROM** EmployeeSalary S **WHERE** E.EmpId = S.EmpId);

**Ques.8. Write a SQL query to fetch duplicate records from a table.**

**SELECT** EmpId, Project, Salary, **COUNT**(\*)

**FROM** EmployeeSalary

**GROUP** **BY** EmpId, Project, Salary

**HAVING** **COUNT**(\*) > 1;

**Ques.9. Write a SQL query to remove duplicates from a table without using temporary table.**

**DELETE** **FROM** EmployeeSalary

**WHERE** EmpId **IN** (

**SELECT** EmpId

**FROM** EmployeeSalary

**GROUP** **BY** Project, Salary

**HAVING** **COUNT**(\*) > 1));

**Ques.10. Write a SQL query to fetch only odd rows from table.**

**SELECT** E.EmpId, E.Project, E.Salary

**FROM** (

**SELECT** \*, Row\_Number() OVER(**ORDER** **BY** EmpId) **AS** RowNumber

**FROM** EmployeeSalary

) E

**WHERE** E.RowNumber % 2 = 1

**Ques.11. Write a SQL query to fetch only even rows from table.**

**SELECT** E.EmpId, E.Project, E.Salary

**FROM** (

**SELECT** \*, Row\_Number() OVER(**ORDER** **BY** EmpId) **AS** RowNumber

**FROM** EmployeeSalary

) E

**WHERE** E.RowNumber % 2 = 0

**Ques.12. Write a SQL query to create a new table with data and structure copied from another table.**

**SELECT** \* **INTO** newTable **FROM** EmployeeDetails;

**Ques.13. Write a SQL query to create an empty table with same structure as some other table.**

Using SELECT INTO command with False 'WHERE' condition-

**SELECT** \* **INTO** newTable **FROM** EmployeeDetails **WHERE** 1 = 0;

This can also done using mySQL 'Like' command with CREATE statement-

**CREATE** **TABLE** newTable **LIKE** EmployeeDetails;

**Ques.14. Write a SQL query to fetch common records between two tables.**

**SELECT** \* **FROM** EmployeeSalary

**INTERSECT**

**SELECT** \* **FROM** ManagerSalary

**Ques.15. Write a SQL query to fetch records that are present in one table but not in another table.**

**SELECT** \* **FROM** EmployeeSalary

**MINUS**

**SELECT** \* **FROM** ManagerSalary

**Ques.16. Write a SQL query to find current date-time.**

mySQL-

**SELECT** NOW();

SQL Server-

**SELECT** getdate();

Oracle- **SELECT** SYSDATE **FROM** DUAL;

**Ques.17. Write a SQL query to fetch all the Employees details from EmployeeDetails table who joined in Year 2016.**

Using BETWEEN for the date range '01-01-2016' AND '31-12-2016'-

**SELECT** \* **FROM** EmployeeDetails

**WHERE** DateOfJoining **BETWEEN** '01-01-2016' **AND** date '31-12-2016';

Also, we can extract year part from the joining date (using YEAR in mySQL)-

**SELECT** \* **FROM** EmployeeDetails

**WHERE** **YEAR**(DateOfJoining) = '2016';

**Ques.18. Write a SQL query to fetch top n records?**

Ans. In mySQL using LIMIT-

**SELECT** \* **FROM** EmployeeSalary **ORDER** **BY** Salary **DESC** **LIMIT** N

In SQL server using TOP command-

**SELECT** TOP N \* **FROM** EmployeeSalary **ORDER** **BY** Salary **DESC**

**Ques.19. Write SQL query to find the nth highest salary from table.**  
Ans. Using Top keyword (SQL Server)-

**SELECT** TOP 1 Salary

**FROM** (

**SELECT** **DISTINCT** TOP N Salary

**FROM** Employee

**ORDER** **BY** Salary **DESC**

)

**ORDER** **BY** Salary **ASC**

Using limit clause(mySQL)

**SELECT** Salary **FROM** Employee **ORDER** **BY** Salary **DESC** **LIMIT** N-1,1;

**Ques.20. Write SQL query to find the 3rd highest salary from table without using TOP/limit keyword.**

**SELECT** Salary

**FROM** EmployeeSalary Emp1

**WHERE** 2 = (

**SELECT** **COUNT**( **DISTINCT** ( Emp2.Salary ) )

**FROM** EmployeeSalary Emp2

**WHERE** Emp2.Salary > Emp1.Salary

)

For nth highest salary-

**SELECT** Salary

**FROM** EmployeeSalary Emp1

**WHERE** N-1 = (

**SELECT** **COUNT**( **DISTINCT** ( Emp2.Salary ) )

**FROM** EmployeeSalary Emp2

**WHERE** Emp2.Salary > Emp1.Salary

)