PRIMARY KEY AND FOREIGN KEY

SQL PRIMARY KEY Constraint

The PRIMARY KEY constraint uniquely identifies each record in a table.

Primary keys must contain UNIQUE values, and cannot contain NULL values.

A table can have only ONE primary key; and in the table, this primary key can consist of single or multiple columns (fields).

SQL FOREIGN KEY Constraint

A FOREIGN KEY is a key used to link two tables together.

A FOREIGN KEY is a field (or collection of fields) in one table that refers to the PRIMARY KEY in another table.

The table containing the foreign key is called the child table, and the table containing the candidate key is called the referenced or parent table.

Look at the following two tables:

Persons

> create table Persons(persionid int not null primary key, name varchar2(20), age int);

personid	name	age
1	Anil	25
2	Mukesh	45
3	Vijay	33

Orders

> create table Orders(orderid int, qty int, personid int,
PRIMARY KEY(orderid) ,
FOREIGN KEY(personid) references Persons(personid));

orderid	qty	personid
10	1000	3
11	1500	1
12	1100	2
13	1100	3

Notice that the "personid" column in the "Orders" table points to the "personid" column in the "Persons" table.

The "personid" column in the "Persons" table is the PRIMARY KEY in the "Persons" table.

The "personid" column in the "Orders" table is a FOREIGN KEY in the "Orders" table.

The FOREIGN KEY constraint is used to prevent actions that would destroy links between tables.

The FOREIGN KEY constraint also prevents invalid data from being inserted into the foreign key column, because it has to be one of the values contained in the table it points to.

1. Retrieve person name, age and order qty from both the tables.
> select Persons.name, Persons.age, Orders.qty from Persons, Orders
where Persons.personid = Orders.personid;

2. Fetch name from Persons table in upper case.
> select upper(name) from Persons;
Note: for lower case: select lower(name) from Persons;

3. Fetch unique values of qty in Orders table.
> select distinct qty from Orders;

4. Find out the length (no. of chars) of each name in the Persons table.
> select length(name) from Persons;

5. Sort the rows of Persons table on name in ascending order.
> select * from Persons order by name asc;
Note: for descending order: select * from Persons order by name desc;

6. Display rows from Persons table where names can be 'Anil' or 'Vijay'.
> select * from Persons where name in ('Anil', 'Vijay');

7. Display rows from Persons table excluding the names 'Anil' or 'Vijay'.
> select * from Persons where name not in ('Anil', 'Vijay');

8. Print details of the person whose name contains 'Muk'.
> select * from Persons where name like '%Muk%';

9. Print details of the person whose name starts with 'An'.
> select * from Persons where name like 'An%';

10. Print details of the person whose name ends with 'ay'.
> select * from Persons where name like '%ay';

11. Display the persons details having exactly 4 letters in the name.
> select * from Persons where name like '____'; // 4 underscores

12. Display the persons details ending with 'y' and contains 5 letters.
> select * from Persons where name like '____y';

13. Display names of the people whose age lies between 30 and 50.
> select name from Persons where age between 30 and 50;

14. How many rows are having qty > 1000 in the Orders table?
> select count(*) from Orders where qty > 1000;

15. Display the number of rows having qty>1000 in the Orders table with an alias name 'NO_RECORDS'.
> select count(*) as NO_RECORDS from Orders where qty > 1000;

16. Display only even rows from Persons table.
> select * from Persons where mod (personid, 2) = 0;
Note: To display odd rows: select * from Persons where mod (personid, 2) <> 0;

Create the following Employee table:

Emp

id	name	salary	dept
10	Anil Kumar	25000.00	Admin
11	Vijay kiran	34500.55	Admin
12	Neel	45000.00	HR
13	Chandra	19000.90	HR
14	Ranjan Kumar	45000.00	Admin
15	Laxmi	33000.00	HR

17. Show highest salary in the table.
> select max(salary) from Emp;
Note: To give highest_salary name for max(salary):
select max(salary) Highest_Salary from Emp;

18. Show second highest salary in the table.
> select max(salary) from Emp
where salary not in (select max(salary) from Emp);

19. Show all depts. along with the number of employees in there.
> select dept, count(dept) from Emp group by dept;

20. Show the last record from Emp table.
> select * from Emp where rowid = (select max(rowid) from Emp);

21. Show the first record from Emp table.
> select * from Emp where rowid = (select min(rowid) from Emp);
Note: select * from Emp where rownum=1;

22. Fetch department names along with total salaries paid for each of them.
> select dept, sum(salary) from Emp group by dept;
Note: to know average salary: avg(salary)

23. Display department along with minimum salary paid in that department.
> select dept, min(salary) from Emp group by dept;

24. Fetch the names and salaries of employees drawing highest salaries.
> select name, salary from Emp where salary = (select max(salary) from Emp);

25. Count how many unique salaries are there in the table.

> select count(distinct salary) from Emp;

26. Display employee row receiving third highest salary.
> select * from Emp a where 3 = (select count(distinct salary) from Emp b
where a.salary <= b.salary);
Note: Apply the above query for 1st highest salary or 2nd highest salary etc.

27. Show employees who are not working in HR dept.
> select * from Emp where dept not in ('HR');

28. Display rows by sorting them on department and on salary in each department.
> select * from Emp order by dept, salary;
Note: to sort on dept in ascending order and salary in descending order: select * from Emp order by dept, salary desc;

29. Display salary, 15% of salary as HRA, 5% of salary as PF.
> select salary, salary*15/100 as HRA, salary*5/100 as PF from Emp;

30. Display rows of employees whose salary is more than that of Laxmi.
> select * from Emp where salary > (select salary from Emp where name = 'Laxmi');

31. Display employee names who earn salary more than that of Laxmi and less than that of Neel.
> select name from Emp where salary>(select salary from Emp where name='Laxmi')
and salary<(select salary from Emp where name='Neel');

32. Display the first 3 letters of the employee names.
> select substr(name, 1,3) from Emp;