```
create table Students(std_id number(4) primary key, std_name varchar(20), std_course
varchar(255), teacher_id number(2));
Create table Teachers(teacher_id number(2) primary key, t_name varchar(25));

Alter table students add constraint fk Foreign key(teacher_id) references Teachers(teacher_id);

Insert into Teachers(teacher_id, t_name)
select 01, 'Sarah' from dual union all
Select 02, 'James' from dual  union all
select 03, 'Lacy' from dual  union all
select 04, 'Thomas' from dual  union all
select 05, 'Harry' from dual;

Insert into Students(std_id, std_name, std_course, teacher_id)
select 1111, 'Mary', 'Physics', 02 from dual union all
select 1112, 'Lucifer', 'Physics', 03 from dual union all
select 1113, 'Taylor', 'Physics', 04 from dual union all
select 1114, 'Adams', 'Physics', 02 from dual union all
select 1115, 'Daisy', 'Physics', 04 from dual union all
select 1116, 'Tom', 'Physics', 02 from dual;

select * from students;
select * from Teachers;


SELECT * FROM STUDENTS,TEACHERS;

select s.std_id ,t.teacher_id from teachers t cross join students s ;
-- INNER JOINS
-- we have to give clauses to identify based onthat things like teacher_id common on both
SELECT S.STD_id,S.STD_NAME,T.T_NAME,T.TEACHER_ID FROM TEACHERS T inner join
STUDENTS S on s.teacher_id = t.teacher_id;

 -- by deafualt ye foreign key fetch krlega and student and teacher ki same row dega like
intersection
SELECT S.STD_id,S.STD_NAME,T.T_NAME,TEACHER_ID FROM TEACHERS T natural join
STUDENTS S;

SELECT S.STD_id,S.STD_NAME,T.T_NAME,TEACHER_ID FROM TEACHERS T join
STUDENTS S using (teacher_id);

-- equal join is the parent of all joins
```

```
-- OUTER JOINS LIKE WORKSON UNIONS

SELECT T.TEACHER_ID,S.STD_ID,T.T_NAME,S.STD_NAME FROM TEACHERS T FULL
OUTER JOIN STUDENTS S ON T.TEACHER_ID = S.TEACHER_ID;

SELECT T.TEACHER_ID,S.STD_ID,T.T_NAME,S.STD_NAME FROM TEACHERS T RIGHT
OUTER JOIN STUDENTS S ON T.TEACHER_ID = S.TEACHER_ID;

-- MEAN TABLE SA PEHLE WALE SAREY COLUMN UTH K AJAYEGE
SELECT T.TEACHER_ID,S.STD_ID,T.T_NAME,S.STD_NAME FROM TEACHERS T LEFT
OUTER JOIN STUDENTS S ON T.TEACHER_ID = S.TEACHER_ID;

SELECT * FROM HR.EMPLOYEES;
SELECT * FROM HR.EMP_DETAILS_VIEW;
SELECT * FROM HR.DEPARTMENTS;

SELECT E1.FIRST_NAME AS EMP_NAME,E2.FIRST_NAME AS MAN_NAME FROM
HR.EMPLOYEES E1 JOIN HR.EMPLOYEES E2 ON E1.EMPLOYEE_ID = E2.MANAGER_ID
ORDER BY E1.EMPLOYEE_ID DESC;

SELECT E.EMPLOYEE_ID,J.EMPLOYEE_ID,J.JOB_ID FROM HR.EMPLOYEES E LEFT
OUTER JOIN HR.JOB_HISTORY J ON E.EMPLOYEE_ID = J.EMPLOYEE_ID;


-- union all also fetch duplicates
-- order,data_types,and No_of columns same hona chaiye
select employee_id,job_id from HR.EMPLOYEES
minus
select employee_id,job_id from hr.job_history;


/*
1. Write a query to list the name, job name, department name,
salary of the employees according to the department in ascending order.
*/
select * from hr.jobs;
select e.first_name||' '||e.last_name as Name,job_title,department_name,salary
  FROM hr.employees e,HR.DEPARTMENTS d,HR.JOBS j
  where e.employee_id = d.department_id and e.job_id = j.job_id ORDER BY
d.DEPARTMENT_ID;

-- Write a query to list the department where at least two employees are working.
SELECT DEPARTMENT_ID FROM HR.EMPLOYEES GROUP BY DEPARTMENT_ID HAVING
```

```
COUNT(*) >=2;


select MANAGER_ID, MIN(SALARY) AS lowest_paid_emp FROM HR.EMPLOYEES
WHERE MANAGER_ID IS NOT NULL GROUP BY MANAGER_ID
HAVING MIN(SALARY) = 2000 ORDER BY MIN(SALARY) DESC;


-- Fetch all the records where salary of employee is less than the lower salary.

SELECT E.EMPLOYEE_ID,E.SALARY,E.FIRST_NAME FROM HR.EMPLOYEES E, HR.JOBS
J WHERE E.JOB_ID = J.JOB_ID AND E.SALARY < J.MIN_SALARY;

/*
Write a query to list the name, job name, annual salary, department id, department name
and city who earn 60000 in a year or not working as an ANALYST.
*/
SELECT * FROM HR.LOCATIONS;
SELECT * FROM HR.DEPARTMENTS;
SELECT E.first_name||' '||E.last_name  AS Name,J.JOB_TITLE,E.SALARY*12 AS
ANNUAL_SALARY,E.DEPARTMENT_ID,D.DEPARTMENT_NAME,
L.CITY FROM HR.EMPLOYEES E JOIN HR.DEPARTMENTS D ON E.DEPARTMENT_ID =
D.DEPARTMENT_ID
JOIN HR.JOBS J ON E.JOB_ID = J.JOB_ID JOIN HR.LOCATIONS L ON D.LOCATION_ID =
L.LOCATION_ID
WHERE (E.SALARY*12)>60000 OR J.JOB_TITLE NOT LIKE 'ANALYST';
```