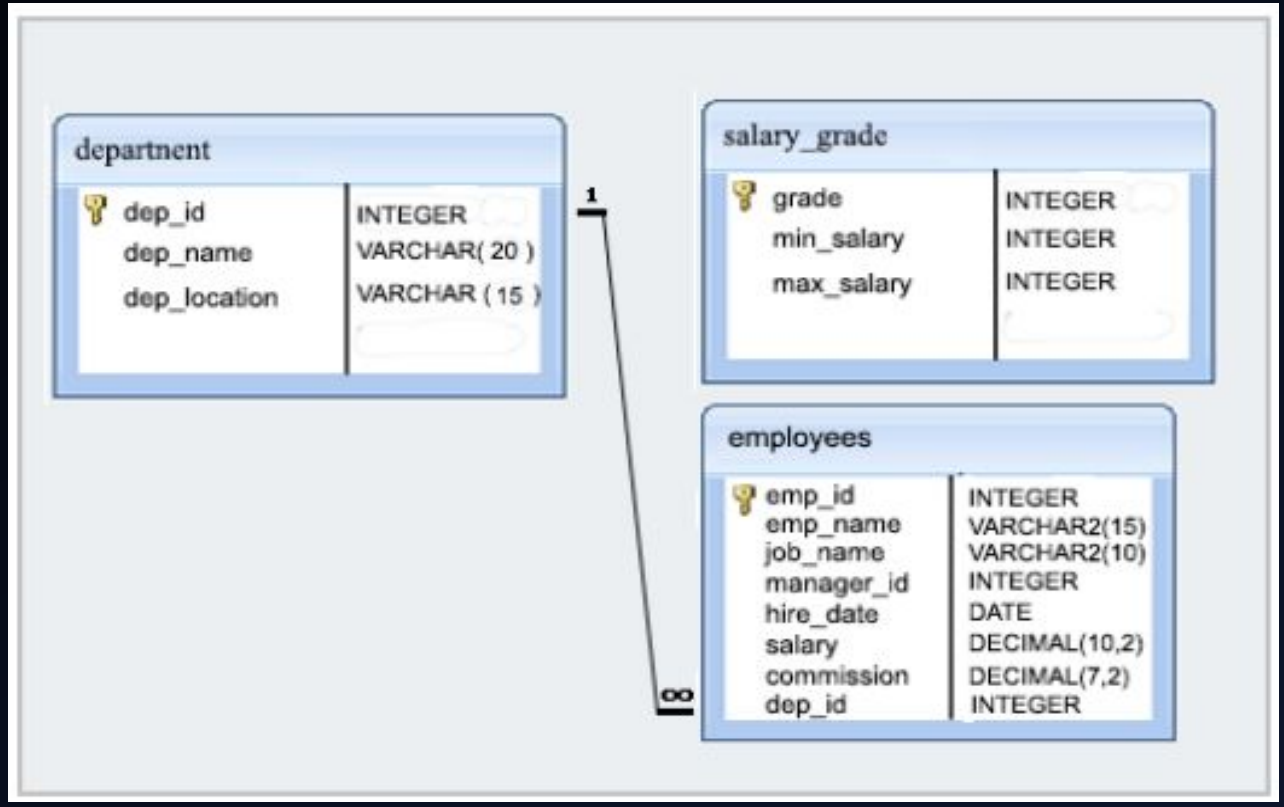
DEV JATINBHAI PATEL

CE076

20CEUOS018

DBMS-LAB11

Create below given Database Schema. Create these relations and find out answer to the queries written below.



```
CREATE TABLE department(
dep_id INTEGER PRIMARY KEY,
dep_name VARCHAR2(20),
dep_location VARCHAR2(15)
);
CREATE TABLE salary_grade(
grade INTEGER PRIMARY KEY,
min_salary INTEGER,
max_salary INTEGER
);
CREATE TABLE employees(
emp_id INTEGER PRIMARY KEY,
emp_name VARCHAR2(15),
job_name VARCHAR2(10),
manager_id INTEGER,
hire_date DATE,
salary DECIMAL(10,2),
commission DECIMAL(7,2),
dep_id INTEGER
);
```

```
Table DEPARTMENT created.


Table SALARY_GRADE created.


Table EMPLOYEES created.
```

Data for employees table:

emp_id	emp_name	job_name	manager_id	hire_date	salary	commission	dep_id
68319	KAYLING	PRESIDENT		1991-11-18	6000		1001
66928	BLAZE	MANAGER	68319	1991-05-01	2750		3001
67832	CLARE	MANAGER	68319	1991-06-09	2550		1001
65646	JONAS	MANAGER	68319	1991-04-02	2957		2001
64989	ADELYN	SALESMAN	66928	1991-02-20	1700	400	3001
65271	WADE	SALESMAN	66928	1991-02-22	1350	600	3001
66564	MADDEN	SALESMAN	66928	1991-09-28	1350	1500	3001
68454	TUCKER	SALESMAN	66928	1991-09-08	1600	0	3001
68736	ADNRES	CLERK	67858	1997-05-23	1200		2001
69000	JULIUS	CLERK	66928	1991-12-03	1050		3001
69324	MARKER	CLERK	67832	1992-01-23	1400		1001
67858	SCARLET	ANALYST	65646	1997-04-19	3100		2001
69062	FRANK	ANALYST	65646	1991-12-03	3100		2001
63679	SANDRINE	CLERK	69062	1990-12-18	900		2001

Data for department table:

dep_id	dep_name	dep_location
1001	FINANCE	SYDNEY
2001	AUDIT	MELBOURNE
3001	MARKETING	PERTH
4001	PRODUCTION	BRISBANE

Data for salary_grade table:

grade	min_sal	max_sal
1	800	1300
2	1301	1500
3	1501	2100
4	2101	3100
5	3101	9999

```
INSERT INTO employees VALUES(68319,'KAYLING','PRESIDENT',null,TO_DATE('1991-11-18','YYYY-MM-DD'),6000,null,1001);

INSERT INTO employees VALUES(66928,'BLAZE','MANAGER',68319,TO_DATE('1991-05-01','YYYY-MM-DD'),2750,null,3001);

INSERT INTO employees VALUES(67832,'CLARE','MANAGER',68319,TO_DATE('1991-06-09','YYYY-MM-DD'),2550,null,1001);

INSERT INTO employees VALUES(65646,'JONAS','MANAGER',68319,TO_DATE('1991-04-02','YYYY-MM-DD'),2957,null,2001);

INSERT INTO employees VALUES(64989,'ADELYN','SALESMAN',66928,TO_DATE('1991-02-20','YYYY-MM-DD'),1700,400,3001);

INSERT INTO employees VALUES(65271,'WADE','SALESMAN',66928,TO_DATE('1991-02-22','YYYY-MM-DD'),1350,600,3001);

INSERT INTO employees
    VALUES(66564,'MADDEN','SALESMAN',66928,TO_DATE('1991-09-28','YYYY-MM-DD'),1350,1500,3001);

INSERT INTO employees VALUES(68454,'TUCKER','SALESMAN',66928,TO_DATE('1991-09-08','YYYY-MM-DD'),1600,null,3001);

INSERT INTO employees VALUES(68736,'ADNRES','CLERK',67858,TO_DATE('1997-05-23','YYYY-MM-DD'),1200,null,2001);

INSERT INTO employees VALUES(69000,'JULIUS','CLERK',66928,TO_DATE('1991-12-03','YYYY-MM-DD'),1050,null,3001);

INSERT INTO employees VALUES(69324,'MARKER','CLERK',67832,TO_DATE('1992-01-23','YYYY-MM-DD'),1400,null,1001);

INSERT INTO employees VALUES(67858,'SCARLET','ANALYST',65646,TO_DATE('1997-04-19','YYYY-MM-DD'),3100,null,2001);

INSERT INTO employees VALUES(69062,'FRANK','ANALYST',65646,TO_DATE('1991-12-03','YYYY-MM-DD'),3100,null,2001);

INSERT INTO employees VALUES(63679,'SANDRINE','CLERK',69062,TO_DATE('1990-12-18','YYYY-MM-DD'),900,null,2001);
```

```
INSERT INTO department VALUES(1001,'FINANCE','SYDNEY');
INSERT INTO department VALUES(2001,'AUDIT','MELBOURNE');
INSERT INTO department VALUES(3001,'MARKETING','PERTH');
INSERT INTO department VALUES(4001,'PRODUCTION','BRISBANE');

INSERT INTO salary_grade VALUES(1,800,1300);
INSERT INTO salary_grade VALUES(2,1301,1500);
INSERT INTO salary_grade VALUES(3,1501,2100);
INSERT INTO salary_grade VALUES(4,2101,3100);
INSERT INTO salary_grade VALUES(5,3101,9999);
```

1. Write a SQL query to find those employees who earn 60000 in a year or not working as an ANALYST. Return employee name, job name, (12*salary) as Annual Salary, department ID, and grade.

A. SELECT emp_name, job_name, (12*salary) as Annual_Salary, dep_id, grade FROM employees, salary_grade WHERE (salary*12>=60000 OR job_name!='ANALYST') AND salary>=min_salary AND salary<=max_salary;

	EMP_NAME	JOB_NAME	ANNUAL_SALARY	DEP_ID	GRADE
1	SANDRINE	CLERK	10800	2001	1
2	JULIUS	CLERK	12600	3001	1
3	ADNRES	CLERK	14400	2001	1
4	WADE	SALESMAN	16200	3001	2
5	MADDEN	SALESMAN	16200	3001	2
6	MARKER	CLERK	16800	1001	2
7	TUCKER	SALESMAN	19200	3001	3
8	ADELYN	SALESMAN	20400	3001	3
9	CLARE	MANAGER	30600	1001	4
10	BLAZE	MANAGER	33000	3001	4
11	JONAS	MANAGER	35484	2001	4
12	KAYLING	PRESIDENT	72000	1001	5

2. Write a SQL query to find those employees who joined before or after the year 1991. Return complete information about the employees.

A. SELECT * FROM employees WHERE EXTRACT(year FROM hire_date)!=1991;

	EMP_ID	EMP_NAME	JOB_NAME	MANAGER_ID	HIRE_DATE	SALARY	COMMISSION	DEP_ID
1	68736	ADNRES	CLERK	67858	23-MAY-97	1200	(null)	2001
2	69324	MARKER	CLERK	67832	23-JAN-92	1400	(null)	1001
3	67858	SCARLET	ANALYST	65646	19-APR-97	3100	(null)	2001
4	63679	SANDRINE	CLERK	69062	18-DEC-90	900	(null)	2001

3. Write a SQL query to find those employees with hire date in the format like February 22, 1991. Return employee ID, employee name, salary, hire date.

A. SELECT emp_id, emp_name, salary, TO_CHAR(hire_date, 'fmMonth DD, YYYY') as hire_date FROM employees;

	EMP_ID	EMP_NAME	SALARY	HIRE_DATE
1	68319	KAYLING	6000	November 18, 1991
2	66928	BLAZE	2750	May 1, 1991
3	67832	CLARE	2550	June 9, 1991
4	65646	JONAS	2957	April 2, 1991
5	64989	ADELYN	1700	February 20, 1991
6	65271	WADE	1350	February 22, 1991
7	66564	MADDEN	1350	September 28, 1991
8	68454	TUCKER	1600	September 8, 1991
9	68736	ADNRES	1200	May 23, 1997
10	69000	JULIUS	1050	December 3, 1991
11	69324	MARKER	1400	January 23, 1992
12	67858	SCARLET	3100	April 19, 1997
13	69062	FRANK	3100	December 3, 1991
14	63679	SANDRINE	900	December 18, 1990

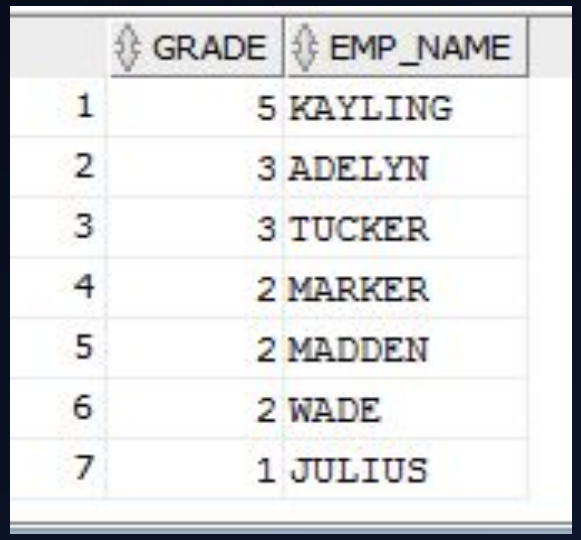
4. Write a SQL query to find those employees whose salary is higher than the salary of their managers. Return employee name, job name, manager ID, salary, manager name, manager's salary.

A. SELECT emp.emp_name, emp.job_name, emp.manager_id, emp.salary, mng.emp_name AS manager_name, mng.salary AS manager_salary FROM employees emp, employees mng WHERE emp.manager_id=mng.emp_id AND emp.salary>mng.salary;

	EMP_NAME	JOB_NAME	MANAGER_ID	SALARY	MANAGER_NAME	MANAGER_SALARY
1	FRANK	ANALYST	65646	3100	JONAS	2957
2	SCARLET	ANALYST	65646	3100	JONAS	2957

5. Write a SQL query to find those employees whose department ID is 1001 or 3001 and salary grade is not 4. They joined the company before 1992-12-31. Return grade, employee name.

A. SELECT grade, emp_name FROM employees, salary_grade WHERE (dep_id=1001 OR dep_id=3001) AND salary>=min_salary AND salary<=max_salary AND grade!=4 AND hire_date<TO_DATE('1992-12-31','YYYY-MM-DD');



	GRADE	EMP_NAME
1	5	KAYLING
2	3	ADELYN
3	3	TUCKER
4	2	MARKER
5	2	MADDEN
6	2	WADE
7	1	JULIUS

6. Write a SQL query to find those employees of MARKETING department come from MELBOURNE or PERTH within the grade 3, 4, and 5 and experience over 25 years. Return department ID, employee ID, employee name, salary, department name, department location and grade.

A. SELECT dep_id, emp_id, emp_name, salary, dep_name, dep_location, grade FROM employees NATURAL JOIN department, salary_grade WHERE dep_name='MARKETING' AND dep_location IN ('PERTH','MELBOURNE') AND grade IN (3,4,5) AND salary>=min_salary AND salary<=max_salary AND (sysdate-hire_date)/365>25;

	DEP_ID	EMP_ID	EMP_NAME	SALARY	DEP_NAME	DEP_LOCATION	GRADE
1	3001	66928	BLAZE	2750	MARKETING	PERTH	4
2	3001	64989	ADELYN	1700	MARKETING	PERTH	3
3	3001	68454	TUCKER	1600	MARKETING	PERTH	3

7. Write a SQL query to find those employees whose grade is 4 and salary between minimum and maximum salary. Return all information of each employees and their grade and salary related details.

A. SELECT * FROM employees WHERE salary BETWEEN (SELECT min_salary FROM salary_grade WHERE grade=4) AND (SELECT max_salary FROM salary_grade WHERE grade=4);

	EMP_ID	EMP_NAME	JOB_NAME	MANAGER_ID	HIRE_DATE	SALARY	COMMISSION	DEP_ID
1	66928	BLAZE	MANAGER	68319	01-MAY-91	2750	(null)	3001
2	67832	CLARE	MANAGER	68319	09-JUN-91	2550	(null)	1001
3	65646	JONAS	MANAGER	68319	02-APR-91	2957	(null)	2001
4	67858	SCARLET	ANALYST	65646	19-APR-97	3100	(null)	2001
5	69062	FRANK	ANALYST	65646	03-DEC-91	3100	(null)	2001

8. Write a SQL query to find the employees in the ascending order of their annual salary. Return employee ID, employee name, monthly salary, salary/30 as Daily_Salary, and 12*salary as Anual_Salary.

A. SELECT emp_id, emp_name, salary, (salary/30) as daily_salary, (12*salary) as annual_salary FROM employees ORDER BY annual_salary ASC;

	EMP_ID	EMP_NAME	SALARY	DAILY_SALARY	ANNUAL_SALARY
1	63679	SANDRINE	900	30	10800
2	69000	JULIUS	1050	35	12600
3	68736	ADNRES	1200	40	14400
4	66564	MADDEN	1350	45	16200
5	65271	WADE	1350	45	16200
6	69324	MARKER	1400	46.666666666666666666666666666666666666667	16800
7	68454	TUCKER	1600	53.333333333333333333333333333333333333333	19200
8	64989	ADELYN	1700	56.666666666666666666666666666666666666667	20400
9	67832	CLARE	2550	85	30600
10	66928	BLAZE	2750	91.666666666666666666666666666666666666667	33000
11	65646	JONAS	2957	98.566666666666666666666666666666666666667	35484
12	69062	FRANK	3100	103.333333333333333333333333333333333333333	37200
13	67858	SCARLET	3100	103.333333333333333333333333333333333333333	37200
14	68319	KAYLING	6000	200	72000