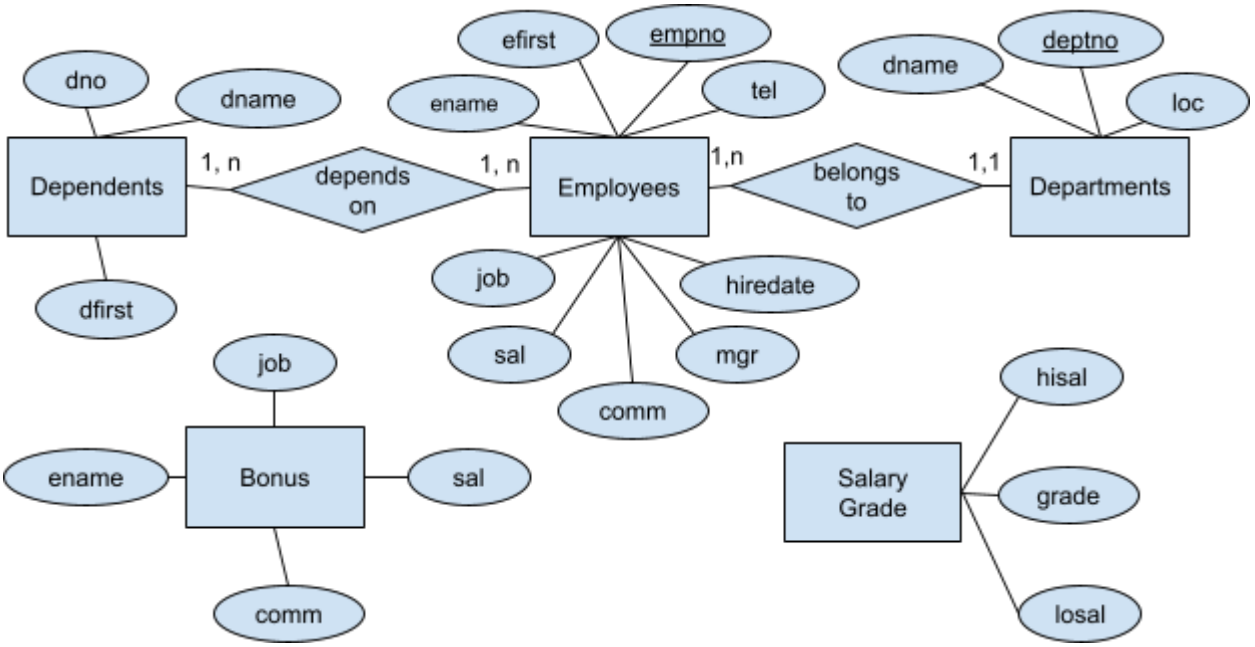
ADV. DB & BIG DATA TP1 REPORT

Exercise 1

1. After creating the database and tables, everything works correctly. Now let's do the exercises
2. Integrity constraints are every primary key (PK) which make every individual unique, also the salary of the employees must be a positive number. Foreign keys (FK) are also a constraint stating that one entity is related to another.

And here's the E/R diagram. We can see all entities and their attributes, along with their relationships:



Relationships:
- 1 Dependant depends on n Employees and 1 Employee has n Dependants. (Many to many). We can note this since the PK of Dependants is a tuple that permits any combination like (1, 2) and (1,3) but also (1,2) and (2,2) where the first one is dno and the second one is empno.
- 1 Employee belongs to 1 department and 1 Department has n Employees. This is because of the foreign key on employees, which allows an employee to belong to an apartment but you can have the same deptno for two employees.

3.

```
ALTER TABLE EMP
ADD CONSTRAINT EFIRST_ENAME_TEL_unique UNIQUE (EFIRST, ENAME, TEL);
```

By doing this we modify the table to add the constraint without needing to drop and create it again. Same applied for next constraints.

4.

```
ALTER TABLE EMP
ADD COLUMN MOBTEL char(10);

ALTER TABLE EMP
ADD CONSTRAINT MOBTEL_check CHECK (MOBTEL LIKE '06%');
```

5.

```
ALTER TABLE EMP
DROP CONSTRAINT fk_emp_dept;
ALTER TABLE EMP
ADD CONSTRAINT fk_emp_dept FOREIGN KEY (DEPTNO) REFERENCES DEPT (DEPTNO) ON DELETE CASCADE;
```

6. Error fixed: Row with first name "KING" has mgr collumn null, which breaks the constraint NOT NULL. To fix it I will put a random value to make it meet the constraint.

```
INSERT INTO EMP VALUES
        (7839, 'KING', 'GUY',  'PRESIDENT', 7839,
        TO_DATE('17-11-1981', 'DD-MM-YYYY'), 5000, NULL, '0149545241',10);
```

7.

```
CREATE SEQUENCE DNO_seq
START WITH 8000
INCREMENT BY 1
MINVALUE 8000
NO CYCLE;
```

8.

```
INSERT INTO DEPENDENTS VALUES (nextval('DNO_seq'), 'JOHNSON', 'MARY', 7369);
INSERT INTO DEPENDENTS VALUES (nextval('DNO_seq'), 'BROWN', 'JAMES', 7876);
INSERT INTO DEPENDENTS VALUES (nextval('DNO_seq'), 'WILLIAMS', 'PATRICIA', 7900);
INSERT INTO DEPENDENTS VALUES (nextval('DNO_seq'), 'JONES', 'MICHAEL', 7934);
INSERT INTO DEPENDENTS VALUES (nextval('DNO_seq'), 'JOHNSON', 'ALICE', 7369);
INSERT INTO DEPENDENTS VALUES (nextval('DNO_seq'), 'WILSON', 'ROBERT', 7900);
INSERT INTO DEPENDENTS VALUES (nextval('DNO_seq'), 'DAVIS', 'EMILY', 7698);
INSERT INTO DEPENDENTS VALUES (nextval('DNO_seq'), 'BROWN', 'MICHAEL', 7876);
INSERT INTO DEPENDENTS VALUES (nextval('DNO_seq'), 'MILLER', 'SARAH', 7788);
```

9.

Following the documentation given at the question, serial is a type which is added instead of “integer” and you can do auto-increment with it, but it has some weird behaviours that makes some things a bit cumbersome, so maybe is not the best option and it's already out of the standard.

Identity is simpler and is in the standard. It's added at creating table too but instead of a type it is an option added as PRIMARY KEY or NOT NULL, etc like: "GENERATED AS IDENTITY".

Sequence is the one used in exercise 7. It's a separated object which increments its value and it can be used when generating data, in the column needed with "nextval(seq_name)".

In my opinion, between identity and sequence, it is better to use identity since it's the standard but also it's simpler, when generating data identity generates the value but with sequence you need to put nextval. Maybe if you need more control with what value you want to be there, sequence could
be better since you can use setval to change the sequence value between more options.

Exercise 2

1.

```
SELECT
    t.table_name,
    array_agg(c.column_name::text) AS COLUMNS
FROM
    information_schema.tables t
INNER JOIN information_schema.columns c ON
    t.table_name = c.table_name
WHERE
    t.table_schema = 'public'
    AND t.table_type= 'BASE TABLE'
    AND c.table_schema = 'public'
GROUP BY t.table name:
```

	table_name name	columns text[]
1	bonus	{sal,job,comm,ename}
2	dependents	{dfirst,dno,dname,empno}
3	dept	{deptno,loc,dname}
4	emp	{ename,efirst,job,mgr,hiredate,sal,comm,tel,deptno,mobtel,empno}
5	project	{pname,budget,projno,startdate}
6	project_emp	{projno,empno}
7	salgrade	{hisal,losal,grade}

2.

```
SELECT * FROM EMP
WHERE comm > sal;
```

	empno [PK] integer	ename character varying (10)	efirst character varying (10)	job character varying (9)	mgr integer	hiredate date	sal integer	comm integer	tel character (10)	deptno integer	mobtel character (10)
1	7654	MARTIN	JOE	SALESMAN	7698	1981-09-28	1250	1400	0149545784	30	[null]

3.

```
SELECT * FROM EMP
WHERE (comm + sal) BETWEEN 1200 AND 2400;
```

	empno [PK] integer	ename character varying (10)	efirst character varying (10)	job character varying (9)	mgr integer	hiredate date	sal integer	comm integer	tel character (10)	deptno integer	mobtel character (10)
1	7499	ALLEN	BOB	SALESMAN	7698	1981-02-20	1600	300	0149547243	30	[null]
2	7521	WARD	PETER	SALESMAN	7698	1981-02-22	1250	500	0149545247	30	[null]
3	7844	TURNER	PETER	SALESMAN	7698	1981-09-08	1500	0	0149548243	30	[null]

4.

```
SELECT * FROM EMP
WHERE job = 'CLERK' OR job = 'ANALYST';
```

	empno [PK] integer	ename character varying (10)	efirst character varying (10)	job character varying (9)	mgr integer	hiredate date	sal integer	comm integer	tel character (10)	deptno integer	mobtel character (10)
1	7369	SMITH	JOHN	CLERK	7902	1980-12-17	800	[null]	0149545243	20	[null]
2	7788	SCOTT	GUY	ANALYST	7566	1982-12-09	3000	[null]	0149545249	20	[null]
3	7876	ADAMS	JOSEPH	CLERK	7788	1983-01-12	1100	[null]	0149565243	20	[null]
4	7900	JAMES	ALAN	CLERK	7698	1981-12-03	950	[null]	0149545564	30	[null]
5	7902	FORD	MARIA	ANALYST	7566	1981-12-03	3000	[null]	0149785243	20	[null]
6	7934	MILLER	ALICE	CLERK	7782	1982-01-23	1300	[null]	0199545243	10	[null]

5.

```
SELECT * FROM EMP
WHERE ename LIKE 'M%';
```

	empno [PK] integer	ename character varying (10)	efirst character varying (10)	job character varying (9)	mgr integer	hiredate date	sal integer	comm integer	tel character (10)	deptno integer	mobtel character (10)
1	7654	MARTIN	JOE	SALESMAN	7698	1981-09-28	1250	1400	0149545784	30	[null]
2	7934	MILLER	ALICE	CLERK	7782	1982-01-23	1300	[null]	0199545243	10	[null]

6.

```
SELECT * FROM EMP
WHERE ename LIKE '_L%';
```

	empno [PK] integer	ename character varying (10)	efirst character varying (10)	job character varying (9)	mgr integer	hiredate date	sal integer	comm integer	tel character (10)	deptno integer	mobtel character (10)
1	7499	ALLEN	BOB	SALESMAN	7698	1981-02-20	1600	300	0149547243	30	[null]
2	7698	BLAKE	BOB	MANAGER	7839	1981-05-01	2850	[null]	0149545254	30	[null]
3	7782	CLARK	JOHN	MANAGER	7839	1981-06-09	2450	[null]	0149545245	10	[null]

7.

```
SELECT * FROM EMP
WHERE (job = 'MANAGER' OR job = 'CLERK')
AND deptno = 10
AND sal > 1500;
```

	empno [PK] integer	ename character varying (10)	efirst character varying (10)	job character varying (9)	mgr integer	hiredate date	sal integer	comm integer	tel character (10)	deptno integer	mobtel character (10)
1	7782	CLARK	JOHN	MANAGER	7839	1981-06-09	2450	[null]	0149545245	10	[null]

8.

```
SELECT * FROM EMP
WHERE comm IS NULL;
```

	empno [PK] integer	ename character varying (10)	efirst character varying (10)	job character varying (9)	mgr integer	hiredate date	sal integer	comm integer	tel character (10)	deptno integer	mobtel character (10)
1	7369	SMITH	JOHN	CLERK	7902	1980-12-17	800	[null]	0149545243	20	[null]
2	7566	JONES	JOHN	MANAGER	7839	1981-04-02	2975	[null]	0149545456	20	[null]
3	7698	BLAKE	BOB	MANAGER	7839	1981-05-01	2850	[null]	0149545254	30	[null]
4	7782	CLARK	JOHN	MANAGER	7839	1981-06-09	2450	[null]	0149545245	10	[null]
5	7788	SCOTT	GUY	ANALYST	7566	1982-12-09	3000	[null]	0149545249	20	[null]
6	7876	ADAMS	JOSEPH	CLERK	7788	1983-01-12	1100	[null]	0149565243	20	[null]
7	7900	JAMES	ALAN	CLERK	7698	1981-12-03	950	[null]	0149545564	30	[null]
8	7902	FORD	MARIA	ANALYST	7566	1981-12-03	3000	[null]	0149785243	20	[null]
9	7934	MILLER	ALICE	CLERK	7782	1982-01-23	1300	[null]	0199545243	10	[null]
10	7839	KING	GUY	PRESIDENT	7839	1981-11-17	5000	[null]	0149545241	10	[null]

9.

```
SELECT * FROM EMP
ORDER BY hiredate ASC;
```

	empno [PK] integer	ename character varying (10)	efirst character varying (10)	job character varying (9)	mgr integer	hiredate date	sal integer	comm integer	tel character (10)	deptno integer	mobtel character (10)
1	7369	SMITH	JOHN	CLERK	7902	1980-12-17	800	[null]	0149545243	20	[null]
2	7499	ALLEN	BOB	SALESMAN	7698	1981-02-20	1600	300	0149547243	30	[null]
3	7521	WARD	PETER	SALESMAN	7698	1981-02-22	1250	500	0149545247	30	[null]
4	7566	JONES	JOHN	MANAGER	7839	1981-04-02	2975	[null]	0149545456	20	[null]
5	7698	BLAKE	BOB	MANAGER	7839	1981-05-01	2850	[null]	0149545254	30	[null]
6	7782	CLARK	JOHN	MANAGER	7839	1981-06-09	2450	[null]	0149545245	10	[null]
7	7844	TURNER	PETER	SALESMAN	7698	1981-09-08	1500	0	0149548243	30	[null]
8	7654	MARTIN	JOE	SALESMAN	7698	1981-09-28	1250	1400	0149545784	30	[null]
9	7839	KING	GUY	PRESIDENT	7839	1981-11-17	5000	[null]	0149545241	10	[null]
10	7900	JAMES	ALAN	CLERK	7698	1981-12-03	950	[null]	0149545564	30	[null]
11	7902	FORD	MARIA	ANALYST	7566	1981-12-03	3000	[null]	0149785243	20	[null]
12	7934	MILLER	ALICE	CLERK	7782	1982-01-23	1300	[null]	0199545243	10	[null]
13	7788	SCOTT	GUY	ANALYST	7566	1982-12-09	3000	[null]	0149545249	20	[null]
14	7876	ADAMS	JOSEPH	CLERK	7788	1983-01-12	1100	[null]	0149565243	20	[null]

10.

```
SELECT * FROM EMP
ORDER BY job, sal DESC;
```

	empno [PK] integer	ename character varying (10)	efirst character varying (10)	job character varying (9)	mgr integer	hiredate date	sal integer	comm integer	tel character (10)	deptno integer	mobtel character (10)
1	7788	SCOTT	GUY	ANALYST	7566	1982-12-09	3000	[null]	0149545249	20	[null]
2	7902	FORD	MARIA	ANALYST	7566	1981-12-03	3000	[null]	0149785243	20	[null]
3	7934	MILLER	ALICE	CLERK	7782	1982-01-23	1300	[null]	0199545243	10	[null]
4	7876	ADAMS	JOSEPH	CLERK	7788	1983-01-12	1100	[null]	0149565243	20	[null]
5	7900	JAMES	ALAN	CLERK	7698	1981-12-03	950	[null]	0149545564	30	[null]
6	7369	SMITH	JOHN	CLERK	7902	1980-12-17	800	[null]	0149545243	20	[null]
7	7566	JONES	JOHN	MANAGER	7839	1981-04-02	2975	[null]	0149545456	20	[null]
8	7698	BLAKE	BOB	MANAGER	7839	1981-05-01	2850	[null]	0149545254	30	[null]
9	7782	CLARK	JOHN	MANAGER	7839	1981-06-09	2450	[null]	0149545245	10	[null]
10	7839	KING	GUY	PRESIDENT	7839	1981-11-17	5000	[null]	0149545241	10	[null]
11	7499	ALLEN	BOB	SALESMAN	7698	1981-02-20	1600	300	0149547243	30	[null]
12	7844	TURNER	PETER	SALESMAN	7698	1981-09-08	1500	0	0149548243	30	[null]
13	7654	MARTIN	JOE	SALESMAN	7698	1981-09-28	1250	1400	0149545784	30	[null]
14	7521	WARD	PETER	SALESMAN	7698	1981-02-22	1250	500	0149545247	30	[null]

11.

```
SELECT * FROM DEPT
WHERE deptno NOT IN (
    SELECT deptno FROM EMP
);
```

	deptno [PK] integer	dname character vary	loc acter varying (13)
1	40	OPERATIONS	BOSTON

Graph Visualiser

12.

```
SELECT e1.empno, e1.ename, e1.efirst, e1.job, e1.mgr, e2.ename AS mgr_name,
       e1.hiredate, e1.sal, e1.comm, e1.tel, e1.deptno, e1.mobtel
FROM EMP e1
LEFT JOIN EMP e2 ON e1.mgr = e2.empno;
```

	empno [PK] integer	ename character varying (10)	efirst character varying (10)	job character varying (9)	mgr integer	mgr_name character varying (10)	hiredate date	sal integer	comm integer	tel character (10)	deptno integer	mobtel character (10)
1	7369	SMITH	JOHN	CLERK	7902	FORD	1980-12-17	800	[null]	0149545243	20	[null]
2	7499	ALLEN	BOB	SALESMAN	7698	BLAKE	1981-02-20	1600	300	0149547243	30	[null]
3	7521	WARD	PETER	SALESMAN	7698	BLAKE	1981-02-22	1250	500	0149545247	30	[null]
4	7566	JONES	JOHN	MANAGER	7839	KING	1981-04-02	2975	[null]	0149545456	20	[null]
5	7654	MARTIN	JOE	SALESMAN	7698	BLAKE	1981-09-28	1250	1400	0149545784	30	[null]
6	7698	BLAKE	BOB	MANAGER	7839	KING	1981-05-01	2850	[null]	0149545254	30	[null]
7	7782	CLARK	JOHN	MANAGER	7839	KING	1981-06-09	2450	[null]	0149545245	10	[null]
8	7788	SCOTT	GUY	ANALYST	7566	JONES	1982-12-09	3000	[null]	0149545249	20	[null]
9	7844	TURNER	PETER	SALESMAN	7698	BLAKE	1981-09-08	1500	0	0149548243	30	[null]
10	7876	ADAMS	JOSEPH	CLERK	7788	SCOTT	1983-01-12	1100	[null]	0149565243	20	[null]
11	7900	JAMES	ALAN	CLERK	7698	BLAKE	1981-12-03	950	[null]	0149545564	30	[null]
12	7902	FORD	MARIA	ANALYST	7566	JONES	1981-12-03	3000	[null]	0149785243	20	[null]
13	7934	MILLER	ALICE	CLERK	7782	CLARK	1982-01-23	1300	[null]	0199545243	10	[null]
14	7839	KING	GUY	PRESIDENT	7839	KING	1981-11-17	5000	[null]	0149545241	10	[null]

13. Since JONES comm is null we can ignore it.

```
SELECT * FROM EMP
WHERE (sal +
    CASE WHEN comm IS NULL THEN 0 ELSE comm END) >
    (SELECT sal FROM EMP WHERE ename = 'JONES');
```

	empno [PK] integer	ename character varying (10)	efirst character varying (10)	job character varying (9)	mgr integer	hiredate date	sal integer	comm integer	tel character (10)	deptno integer	mobtel character (10)
1	7788	SCOTT	GUY	ANALYST	7566	1982-12-09	3000	[null]	0149545249	20	[null]
2	7902	FORD	MARIA	ANALYST	7566	1981-12-03	3000	[null]	0149785243	20	[null]
3	7839	KING	GUY	PRESIDENT	7839	1981-11-17	5000	[null]	0149545241	10	[null]

14.

```
SELECT empno, ename, efirst, job, mgr, hiredate,
    (sal, CASE WHEN comm IS NULL THEN 0 ELSE comm END) AS earning,
    tel, deptno, mobtel FROM EMP
```

	empno [PK] integer	ename character varying (10)	efirst character varying (10)	job character varying (9)	mgr integer	hiredate date	earning record	tel character (10)	deptno integer
1	7369	SMITH	JOHN	CLERK	7902	1980-12-17	800,0	0149545243	20
2	7499	ALLEN	BOB	SALESMAN	7698	1981-02-20	1600,300	0149547243	30
3	7521	WARD	PETER	SALESMAN	7698	1981-02-22	1250,500	0149545247	30
4	7566	JONES	JOHN	MANAGER	7839	1981-04-02	2975,0	0149545456	20
5	7654	MARTIN	JOE	SALESMAN	7698	1981-09-28	1250,1400	0149545784	30
6	7698	BLAKE	BOB	MANAGER	7839	1981-05-01	2850,0	0149545254	30
7	7782	CLARK	JOHN	MANAGER	7839	1981-06-09	2450,0	0149545245	10
8	7788	SCOTT	GUY	ANALYST	7566	1982-12-09	3000,0	0149545249	20
9	7844	TURNER	PETER	SALESMAN	7698	1981-09-08	1500,0	0149548243	30
10	7876	ADAMS	JOSEPH	CLERK	7788	1983-01-12	1100,0	0149565243	20
11	7900	JAMES	ALAN	CLERK	7698	1981-12-03	950,0	0149545564	30
12	7902	FORD	MARIA	ANALYST	7566	1981-12-03	3000,0	0149785243	20
13	7934	MILLER	ALICE	CLERK	7782	1982-01-23	1300,0	0199545243	10
14	7839	KING	GUY	PRESIDENT	7839	1981-11-17	5000,0	0149545241	10

15.

```
SELECT deptno FROM DEPT
WHERE deptno IN (SELECT deptno FROM EMP);
```

	deptno [PK] integer
1	10
2	20
3	30

16.

```
SELECT e.empno, e.ename, e.efirst, e.job, e.mgr, e.hiredate, e.sal,
       e.comm, e.tel, e.deptno, e.mobtel
FROM EMP e JOIN DEPT d on e.deptno = d.deptno
WHERE d.loc = 'CHICAGO'
AND e.job = (
    SELECT job FROM EMP
    WHERE ename = 'JONES'
```

	empno [PK] integer	ename character varying (10)	efirst character varying (10)	job character varying (9)	mgr integer	hiredate date	sal integer	comm integer	tel character (10)	deptno integer	mobtel character (10)
1	7698	BLAKE	BOB	MANAGER	7839	1981-05-01	2850	[null]	0149545254	30	[null]

17.

```
SELECT e1.empno, e1.ename, e1.efirst, e1.job, e1.mgr, e1.hiredate,
       e1.sal, e1.comm, e1.tel, e1.deptno, e1.mobtel
FROM EMP e1 JOIN EMP e2 ON e1.mgr = e2.empno
WHERE e1.deptno != e2.deptno;
```

	empno [PK] integer	ename character varying (10)	efirst character varying (10)	job character varying (9)	mgr integer	hiredate date	sal integer	comm integer	tel character (10)	deptno integer	mobtel character (10)
1	7566	JONES	JOHN	MANAGER	7839	1981-04-02	2975	[null]	0149545456	20	[null]
2	7698	BLAKE	BOB	MANAGER	7839	1981-05-01	2850	[null]	0149545254	30	[null]

18.

```
SELECT * FROM EMP
WHERE deptno IN (
    SELECT deptno FROM EMP
    WHERE job = 'CLERK'
);
```

	empno [PK] integer	ename character varying (10)	efirst character varying (10)	job character varying (9)	mgr integer	hiredate date	sal integer	comm integer	tel character (10)	deptno integer	mobtel character (10)
1	7369	SMITH	JOHN	CLERK	7902	1980-12-17	800	[null]	0149545243	20	[null]
2	7499	ALLEN	BOB	SALESMAN	7698	1981-02-20	1600	300	0149547243	30	[null]
3	7521	WARD	PETER	SALESMAN	7698	1981-02-22	1250	500	0149545247	30	[null]
4	7566	JONES	JOHN	MANAGER	7839	1981-04-02	2975	[null]	0149545456	20	[null]
5	7654	MARTIN	JOE	SALESMAN	7698	1981-09-28	1250	1400	0149545784	30	[null]
6	7698	BLAKE	BOB	MANAGER	7839	1981-05-01	2850	[null]	0149545254	30	[null]
7	7782	CLARK	JOHN	MANAGER	7839	1981-06-09	2450	[null]	0149545245	10	[null]
8	7788	SCOTT	GUY	ANALYST	7566	1982-12-09	3000	[null]	0149545249	20	[null]
9	7844	TURNER	PETER	SALESMAN	7698	1981-09-08	1500	0	0149548243	30	[null]
10	7876	ADAMS	JOSEPH	CLERK	7788	1983-01-12	1100	[null]	0149565243	20	[null]
11	7900	JAMES	ALAN	CLERK	7698	1981-12-03	950	[null]	0149545564	30	[null]
12	7902	FORD	MARIA	ANALYST	7566	1981-12-03	3000	[null]	0149785243	20	[null]
13	7934	MILLER	ALICE	CLERK	7782	1982-01-23	1300	[null]	0199545243	10	[null]
14	7839	KING	GUY	PRESIDENT	7839	1981-11-17	5000	[null]	0149545241	10	[null]

19.

```
SELECT * FROM EMP
WHERE deptno = 10 AND job IN (
    SELECT job FROM EMP
    WHERE deptno IN (
        SELECT deptno FROM DEPT
        WHERE dname = 'SALES'
    )
);
```

	empno [PK] integer	ename character varying (10)	efirst character varying (10)	job character varying (9)	mgr integer	hiredate date	sal integer	comm integer	tel character (10)	deptno integer	mobtel character (10)
1	7782	CLARK	JOHN	MANAGER	7839	1981-06-09	2450	[null]	0149545245	10	[null]
2	7934	MILLER	ALICE	CLERK	7782	1982-01-23	1300	[null]	0199545243	10	[null]

20.

```sql
SELECT * FROM EMP
WHERE job IN (
    SELECT job FROM EMP
    WHERE ename = 'JONES'
)
UNION
SELECT * FROM EMP
WHERE sal > (
    SELECT sal FROM EMP
    WHERE ename = 'FORD'
);
```

	empno integer	ename character varying (10)	efirst character varying (10)	job character varying (9)	mgr integer	hiredate date	sal integer	comm integer	tel character (10)	deptno integer	mobtel character (10)
1	7566	JONES	JOHN	MANAGER	7839	1981-04-02	2975	[null]	0149545456	20	[null]
2	7698	BLAKE	BOB	MANAGER	7839	1981-05-01	2850	[null]	0149545254	30	[null]
3	7839	KING	GUY	PRESIDENT	7839	1981-11-17	5000	[null]	0149545241	10	[null]
4	7782	CLARK	JOHN	MANAGER	7839	1981-06-09	2450	[null]	0149545245	10	[null]

21.

```sql
SELECT * FROM EMP
WHERE sal > ALL (
    SELECT sal FROM EMP
    WHERE deptno = 20
)
```

	empno [PK] integer	ename character varying (10)	efirst character varying (10)	job character varying (9)	mgr integer	hiredate date	sal integer	comm integer	tel character (10)	deptno integer	mobtel character (10)
1	7839	KING	GUY	PRESIDENT	7839	1981-11-17	5000	[null]	0149545241	10	[null]

Exercise 3

1.

```sql
CREATE TABLE IF NOT EXISTS PROJECT (
    PROJNO integer constraint pk_project primary key,
    PNAME varchar(20),
    STARTDATE DATE,
    BUDGET integer
);
```

2.

```
CREATE TABLE IF NOT EXISTS PROJECT_EMP (
    EMPNO integer,
    PROJNO integer,
    constraint pk_project_emp primary key (EMPNO, PROJNO),
    constraint fk_emp foreign key (EMPNO) references EMP (EMPNO),
    constraint fk_project foreign key (PROJNO) references PROJECT (PROJNO)
);
```

```
INSERT INTO PROJECT VALUES (1, 'INNOVATION', TO_DATE('15-01-2023', 'DD-MM-YYYY'), 50000);
INSERT INTO PROJECT VALUES (2, 'RESEARCH', TO_DATE('12-07-2023 ', 'DD-MM-YYYY'), 80000);
INSERT INTO PROJECT VALUES (3, 'MARKETINGCAMPAIGNS', TO_DATE('04-01-2024', 'DD-MM-YYYY'), 75000);
INSERT INTO PROJECT VALUES (4, 'FINANTIALAUDITS', TO_DATE('15-03-2024', 'DD-MM-YYYY'), 25000);

INSERT INTO PROJECT_EMP VALUES (7566, 1);
INSERT INTO PROJECT_EMP VALUES (7566, 2);
INSERT INTO PROJECT_EMP VALUES (7566, 3);
INSERT INTO PROJECT_EMP VALUES (7566, 4);
INSERT INTO PROJECT_EMP VALUES (7902, 4);
INSERT INTO PROJECT_EMP VALUES (7499, 1);
INSERT INTO PROJECT_EMP VALUES (7369, 2);
INSERT INTO PROJECT_EMP VALUES (7521, 3);
INSERT INTO PROJECT_EMP VALUES (7934, 4);
INSERT INTO PROJECT_EMP VALUES (7839, 2);
INSERT INTO PROJECT_EMP VALUES (7839, 1);
INSERT INTO PROJECT_EMP VALUES (7788, 1);
INSERT INTO PROJECT_EMP VALUES (7902, 1);
INSERT INTO PROJECT_EMP VALUES (7782, 2);
INSERT INTO PROJECT_EMP VALUES (7844, 2);
INSERT INTO PROJECT_EMP VALUES (7654, 3);
INSERT INTO PROJECT_EMP VALUES (7900, 3);
INSERT INTO PROJECT_EMP VALUES (7698, 4);
INSERT INTO PROJECT_EMP VALUES (7782, 4);
INSERT INTO PROJECT_EMP VALUES (7876, 4);
```

3.

```
SELECT empno FROM PROJECT_EMP
GROUP BY empno
HAVING COUNT(projno) = (SELECT COUNT(*) FROM PROJECT);
```

	empno integer
1	7566

4. The view represents information (empno, ename and deptno) from employees who work at sales department (deptno = 10). WITH CHECK OPTION ensures that any addition to this view should meet the condition of the view, in this case, deptno should be 10.

```
CREATE VIEW sales_staff AS

SELECT empno, ename, deptno

    FROM emp

    WHERE deptno = 10 WITH CHECK OPTION
```

5. It is expected that since Williams is at department 30 throws an error which says something about the condition, but it seems it's trying to add those two employees to the EMP table too. However, since there is just 3 data from those 2 employees and we remember at least mgr should'n be null with our configuration, that's why i get a different error, which says mgr is null. (Sorry that the error is in Spanish but it's basically what I'm explaining).

```
NOTICE:  la relación «project» ya existe, omitiendo
NOTICE:  la relación «project_emp» ya existe, omitiendo

ERROR:  La fila que falla contiene (7584, OSTER, null, null, null, null, null, null, null, 10, null).el valor nulo en la columna «mgr» de la relación «emp» viola la restricción de
no nulo

ERROR:  el valor nulo en la columna «mgr» de la relación «emp» viola la restricción de no nulo
SQL state: 23502
Detail: La fila que falla contiene (7584, OSTER, null, null, null, null, null, null, null, 10, null).
```

6.

```
SELECT empno, COUNT(projno) FROM PROJECT_EMP
GROUP BY empno
HAVING COUNT(projno) >= 2;
```

	empno integer	count bigint
1	7839	2
2	7902	2
3	7782	2
4	7566	4

7.

```sql
SELECT empno FROM PROJECT_EMP
WHERE projno = 1
INTERSECT
SELECT empno FROM PROJECT_EMP
WHERE projno = 2;

SELECT empno FROM PROJECT_EMP
WHERE projno = 3
EXCEPT
SELECT empno FROM PROJECT_EMP
WHERE projno = 4;
```

	empno integer
1	7839
2	7566

	empno integer
1	7521
2	7900
3	7654

8.

```sql
SELECT empno, projno, sal
FROM (
  SELECT empno, projno, sal
  FROM PROJECT_EMP pe1 NATURAL JOIN EMP e1
  WHERE (
    SELECT COUNT(*)
    FROM PROJECT_EMP pe2 NATURAL JOIN EMP e2
    WHERE pe2.projno = pe1.projno AND e2.sal > e1.sal
  ) < 3
) AS TopEmployees
ORDER BY projno, sal DESC;
```

	empno integer	projno integer	sal integer
1	7839	1	5000
2	7902	1	3000
3	7788	1	3000
4	7839	2	5000
5	7566	2	2975
6	7782	2	2450
7	7566	3	2975
8	7521	3	1250
9	7654	3	1250
10	7902	4	3000
11	7566	4	2975
12	7698	4	2850

9.

.1

```
SELECT empno,
    ROUND(COUNT(projno) * 100/(SELECT COUNT(*)
    FROM PROJECT), 2) AS percentage FROM PROJECT_EMP
GROUP BY empno
```

	empno integer	percentage numeric
1	7839	50.00
2	7902	50.00
3	7698	25.00
4	7369	25.00
5	7499	25.00
6	7900	25.00
7	7788	25.00
8	7876	25.00
9	7782	50.00
10	7844	25.00
11	7934	25.00
12	7566	100.00
13	7521	25.00
14	7654	25.00

.2

```sql
SELECT empno, percentage,
       CASE
           WHEN percentage = 0 THEN 'Empty'
           WHEN percentage BETWEEN 10 AND 49 THEN 'Small'
           WHEN percentage BETWEEN 50 AND 79 THEN 'Medium'
           WHEN percentage BETWEEN 80 AND 99 THEN 'Large'
           WHEN percentage = 100 THEN 'Total'
       END AS scope_size
FROM (
    SELECT empno,
           ROUND(COUNT(projno) * 100.0 / (SELECT COUNT(*)
           FROM PROJECT), 2) AS percentage
    FROM PROJECT_EMP
    GROUP BY empno
);
```

	empno integer	percentage numeric	scope_size text
1	7839	50.00	Medium
2	7902	50.00	Medium
3	7698	25.00	Small
4	7369	25.00	Small
5	7499	25.00	Small
6	7900	25.00	Small
7	7788	25.00	Small
8	7876	25.00	Small
9	7782	50.00	Medium
10	7844	25.00	Small
11	7934	25.00	Small
12	7566	100.00	Total
13	7521	25.00	Small
14	7654	25.00	Small