1 - CREAR UNA CONSULTA DE UNION QUE SELECCIONE LOS CAMPOS EMPLOYEE\_ID, JOB\_ID, Y DEPARTMENT\_ID DE LAS TABLAS EMPLOYEES Y JOB\_HISTORY

SELECT employee\_id, job\_id, department\_id

FROM employees

UNION

SELECT employee\_id, job\_id, department\_id

FROM job\_history;

2 - CREAR UNA CONSULTA DE INTERSECCION QUE SELECCIONE LOS CAMPOS EMPLOYEE\_ID, JOB\_ID, Y DEPARTMENT\_ID DE LAS TABLAS EMPLOYEES Y JOB\_HISTORY

SELECT employee\_id, job\_id, department\_id

FROM employees

INTERSECT

SELECT employee\_id, job\_id, department\_id

FROM job\_history;

3 - CREAR UNA CONSULTA QUE SELECCIONE LOS CAMPOS EMPLOYEE\_ID, JOB\_ID, Y DEPARTMENT\_ID DE LA TABLA EMPLOYEES, MENOS(OPERADOR MINUS) LOS CAMPOS EMPLOYEE\_ID, JOB\_ID, Y DEPARTMENT\_ID DE LA TABLA JOB\_HISTORY

SELECT employee\_id,job\_id, department\_id

FROM employees

MINUS

SELECT employee\_id,job\_id,department\_id

FROM job\_history;

4 - CREAR UNA CONSULTA DE CORRESPONDENCIA QUE CONTENGA LOS CAMPOS DEPARTMENT\_ID, JOB\_ID Y JOB\_TITLE DE LAS TABLAS EMPLOYEES Y JOBS

SELECT job\_id, NULL AS JOB\_TITLE, hire\_date

FROM employees

UNION

SELECT job\_id, job\_title,NULL

FROM jobs;

5 - CREAR UNA CONSULTA DE CONTROL DE ORDEN DE FILAS DE 4 FILAS UTILIZANDO LA TABLA DUAL

SELECT 'Oracle' AS My\_dream, 3 a\_dummy

FROM dual

UNION

SELECT 'Sistemas integrados', 1 a\_dummy

FROM dual

UNION

SELECT 'IT', 2 a\_dummy

FROM dual

UNION

SELECT 'Visual Basic', 4 a\_dummy

FROM dual

ORDER BY a\_dummy;

6 - CREAR UNA CONSULTA DE DESPLAZAMIENTO POR ARBOL DE LA TABLA EMPLOYEES DE ARRIBA HACIA ABAJO, QUE SELECCIONE LOS CAMPOS EMPLOYEE\_ID, LAST\_NAME, FIRST\_NAME, SALARY, MANAGER\_ID, LEVEL

SELECT employee\_id, last\_name, first\_name, salary, manager\_id, LEVEL

FROM employees

START WITH employee\_id = 100

CONNECT BY PRIOR employee\_id = manager\_id;

7 - CREAR UNA CONSULTA DE DESPLAZAMIENTO POR ARBOL DE LA TABLA EMPLOYEES DE ABAJO HACIA ARRIBA, QUE SELECCIONE LOS CAMPOS EMPLOYEE\_ID, LAST\_NAME, JOB\_ID, SALARY QUE COMIENCE CON EL MAXIMO EMPLOYEE\_ID

SELECT employee\_id, last\_name, job\_id, manager\_id, salary

FROM employees

START WITH employee\_id =( SELECT MAX(EMPLOYEE\_ID) FROM EMPLOYEES)

CONNECT BY PRIOR manager\_id = employee\_id;

8 - CREAR UNA CONSULTA DE CLASIFICACIÓN DE FILAS CON LA PSEUDOCOLUMNA LEVEL, QUE SELECCIONE APELLIDO Y NOMBRES DE LA TABLA EMPLOYEES QUE COMIENCE CON EL APELLIDO King

SELECT LPAD(last\_name||' '||first\_name, LENGTH(last\_name||' '||first\_name)+(LEVEL\*2)-2,'\_')

AS org\_chart

FROM employees

START WITH last\_name='King'

CONNECT BY PRIOR employee\_id=manager\_id;

9 - PARTIENDO DE LA RAIZ, DESPLACESE DE ARRIBA HACIA ABAJO Y ELIMINE EL EMPLEADO KING DEL RESULTADO, PERO PROCESE LAS FILAS SECUNDARIAS, MOSTRANDO LAS COLUMNAS DEPARTMENT\_ID, EMPLOYEE\_ID, LAST\_NAME, JOB\_ID, SALARY DE LA TABLA EMPLOYEES.

SELECT department\_id, employee\_id,last\_name, job\_id, salary

FROM employees

WHERE last\_name != 'King'

START WITH manager\_id = (select min(manager\_id) from employees)

CONNECT BY PRIOR employee\_id = manager\_id;

10 - PARTIENDO DE LA RAIZ, DESPLACESE DE ARRIBA ABAJO Y ELIMINE EL EMPLEADO Greenberg DEL RESULTADO Y TODAS LAS FILAS SECUNDARIAS, MOSTRANDO LAS COLUMNAS DEPARTMENT\_ID, EMPLOYEE\_ID, LAST\_NAME, JOB\_ID, SALARY DE LA TABLA EMPLOYEES.

SELECT department\_id, employee\_id, last\_name, job\_id, salary

FROM employees

START WITH manager\_id IS NULL

CONNECT BY PRIOR employee\_id = manager\_id

AND last\_name != 'Greenberg';

11 - CREE UNA CONSULTA QUE MUESTRE LOS APELLIDOS QUE CONTENGAN LETRAS 'A' Y 'b' DE LA TABLA EMPLOYEES

SELECT last\_name

FROM employees

WHERE REGEXP\_LIKE (last\_name, '(A|b)');

12 - CREE UNA CONSULTA QUE EXTRAIGA LOS PUNTOS (.) DEL CAMPO PHONE\_NUMBER DE LA TABLA EMPLOYEES

SELECT

REGEXP\_REPLACE(phone\_number,

'([[:digit:]]{3})\.([[:digit:]]{3})\.([[:digit:]]{4})',

'\1\2\3') REGEXP\_REPLACE

FROM employees;

13 - CREE UNA TABLA DENOMINADA ARTICULO QUE CONTENGA LOS CAMPOS COD\_ART NUMBER, DESCRIPCION VARCHAR2(50), QUE EL CAMPO PK SEA COD\_ART, Y QUE EL CAMPO DESCRIPCION NO PERMITA VALORES NULOS.

CREATE TABLE ARTICULO(COD\_ART NUMBER PRIMARY KEY NOT NULL,

DESCRIPCION VARCHAR2(50) NOT NULL);

14 - CREE UNA CONSTRAINT PARA EL CAMPO EMAIL DE LA TABLA EMPLOYEES, QUE VALIDE QUE DICHO CAMPO CONTENGA EL CARACTER @.

ALTER TABLE EMPLOYEES

ADD CONSTRAINT email\_addr

CHECK(REGEXP\_LIKE(email,'@'))NOVALIDATE ;

15 - CREE UNA CONSTRAINT QUE NO PERMITA VALORES NULOS EN EL CAMPO PHONE NUMBER DE LA TABLA EMPLOYEES

alter table EMPLOYEES modify PHONE\_NUMBER not null NOVALIDATE;