BASE DE DATOS

MARIANA ANDRADE GARCIA UP200651

DIEGO SANCHEZ OLVERA UP210010

OBET LEONARDO FUENTES ONTIVEROS UP200677

ANGEL IBARRA VIVEROS UP200635

HECTOR ENRIQUE AGUAYO GARCIA UP200428

CRISTOPHER KALEB RUIZ ORTIZ UP200674

OSCAR ANTONIO CHAVEZ CASTILLO 200770

GONZALO GUADALUPE GONZALEZ ALBA UP200420

IXCHEL ALEJANDRA FLORES ANDRADE UP 200417

RAFAEL MARTINEZ VERDIN UP200190

ISC06A

FECHA DE ENTREGA: 20 DE NOVIEMBRE DEL 2022

1. Cree las tablas adicionales que se utilizan en esta sección ejecutando las siguientes sentencias:

```
CREATE TABLE emp AS SELECT * FROM employees;

CREATE TABLE dept AS SELECT * FROM departments;
```

2. Cree un informe que muestre el nombre de restricción, el tipo, el nombre de columna y la posición de columna de todas las restricciones de la tabla JOB_HISTORY, además de las restricciones no nulas.

```
CREATE TABLE my_temp_table AS

(SELECT cons.constraint_name, cons.constraint_type,cols.column_name, cols.position,

TO_LOB(cons.search_condition) search_condition FROM user_constraints cons

INNER JOIN user_cons_columns cols

ON cons.constraint_name = cols.constraint_name

WHERE cons.table_name = 'JOB_HISTORY' );
```

3. Cree una restricción de clave primaria en la columna employee_id de la tabla emp.

```
ALTER TABLE emp

ADD CONSTRAINT emp_employee_id_pk PRIMARY KEY (employee_id);
```

4. Cree una clave primaria en la columna department_id de la tabla dept.

```
ALTER TABLE dept

ADD CONSTRAINT dept_department_id_pk PRIMARY KEY (department_id);
```

5. Agregue una restricción ajena entre DEPT y EMP, de modo que solo se puedan introducir departamentos válidos en la tabla EMP. Asegúrese de que puede suprimir cualquier fila de la tabla DEPT y de que se suprimen las filas a las que se hace referencia en la tabla EMP.

```
ALTER TABLE emp ADD CONSTRAINT emp_dept_department_id_fk FOREIGN

KEY (department_id)

REFERENCES dept (department_id) ON DELETE CASCADE;
```

6. Pruebe la restricción de clave ajena que acaba de crear: Cuente el número de filas en la tabla EMP.

SELECT COUNT(*) FROM emp;

Elimine el departamento 10 de la tabla dept.

DELETE FROM dept WHERE department_id = 10;

Ahora vuelva a contar los empleados. Debería haber menos empleados.

SELECT COUNT(*) FROM emp;

7. Genere un informe que devuelva el apellido, el salario, el número de departamento y el salario medio de todos los departamentos en los que el salario es mayor que el salario medio.

8. Cree una vista denominada V2 que devuelva el salario más alto, el salario más bajo, el salario medio y el nombre del departamento.

```
CREATE OR REPLACE VIEW v2 ("highest salary", "lowest salary", "average salary",
"Department Name") AS
SELECT
TO_CHAR(ROUND(MAX(NVL(emp.salary,0)),2),'$999999.99'),
TO_CHAR(ROUND(MIN(NVL(emp.salary,0)),2),'$999999.99'),
TO_CHAR(ROUND(AVG(NVL(emp.salary,0)),2),'$999999.99'), dpt.department_name
FROM departments dpt LEFT OUTER JOIN employees emp ON dpt.department_id =
emp.department_id
GROUP BY (dpt.department_id, dpt.department_name);
```

9. Cree una vista denominada Dept_Managers_view que devuelva una lista de nombres de departamento junto con las iniciales y el apellido del jefe para dicho departamento. Pruebe la vista devolviendo todas sus filas. Asegúrese de que no se pueda actualizar ninguna fila a través de la vista. Pruebe a ejecutar una sentencia UPDATE en la vista.

```
CREATE OR REPLACE VIEW dept_managers_view AS
SELECT DISTINCT SUBSTR(NVL(mgr.first_name, '_'),1, 1) ||
```

SUBSTR(mgr.last_name,1, 1) initials, mgr.last_name surname, dpt.department_name

FROM

employees mgr INNER JOIN employees emp ON mgr.employee_id = emp.manager_id

LEFT OUTER JOIN departments dpt ON mgr.department_id = dpt.department_id;

10. Cree una secuencia denominada ct_seq con todos los valores por defecto.

CREATE SEQUENCE ct_seq ;

11. Examine la siguiente sentencia de inserción y corrija los errores.

INSERT INTO emp

(employee_id, first_name, last_name, email, phone_number, hire_date,

job_id, salary, commission_pct, manager_id, department_id)

VALUES

(ct_seq.nextvalue, "Kaare", 'Hansen', 'KHANSEN', '44965 832123',

sysdate, 'SA_REP', $6500, null, 100, 20);

INSERT INTO emp

(employee_id, first_name, last_name, email, phone_number,

hire_date,

job_id, salary, commission_pct, manager_id, department_id)

VALUES

(ct_seq.NEXTVAL, 'Kaare', 'Hansen', 'KHANSEN', '44965 832123',

sysdate, 'SA_REP', 6500, null, 100, 20);

12. Escriba la sentencia SQL para mostrar todas las tablas de usuario que contienen el

nombre PRIV.

SELECT * FROM all_tables WHERE REGEXP_LIKE(table_name, '(PRIV)');

13. Conceda acceso de selección a público en la tabla EMP y verifique que se ha

otorgado mediante la ejecución esta consulta.

```
SELECT *
FROM user_tab_privs
WHERE table_name = 'EMP';
GRANT SELECT ON emp to PUBLIC;
```

14. Sustituya ?? en la siguiente consulta mediante expresiones regulares para devolver solo los números de la siguiente cadena: 'Oracle Academy9547d6905%&^ db apex'.

```
SELECT REGEXP_REPLACE('Oracle Academy9547d6905%&^ db apex',??,'')
regexpreplace
FROM DUAL;
SELECT REGEXP_REPLACE('Oracle Academy9547d6905%&^ db apex','[^0-
9]','') regexpreplace
FROM DUAL;
```

15. Corrija la consulta anterior mediante expresiones regulares para devolver el número de dígitos de la siguiente cadena: 'Oracle Academy9547d6905 %y;^ db'

```
SELECT LENGTH(REGEXP_REPLACE('Oracle Academy9547d6905%&^ db
apex','??','')) regexpreplace
FROM DUAL;
SELECT LENGTH(REGEXP_REPLACE('Oracle Academy9547d6905%&^ db
apex','[^[:digit:]]','')) regexpreplace
FROM DUAL;
```

16. Corrija la consulta de nuevo para devolver solo los caracteres no numéricos.

```
SELECT REGEXP_REPLACE('Oracle Academy9547d6905%&^ db apex','??','')
regexpreplace
FROM DUAL;
SELECT REGEXP_REPLACE('Oracle Academy9547d6905%&^ db apex','[[:digit:]]','')
regexpreplace
```

FROM DUAL;

17. Mediante las uniones propiedad de Oracle, construya una instrucción que devuelva todos los employee_ids unidos a todos los department_names.

```
SELECT em.employee_id, dp.department_name
FROM employees em, departments dp;
```

18. Vuelva a utilizar las uniones Oracle para corregir la sentencia anterior de modo que devuelva solo el nombre del departamento en el que está trabajando el empleado actualmente.

```
SELECT em.employee_id, dp.department_name
FROM employees em, departments dp
WHERE em.department_id = dp.department_id;
```

19. Vuelva a utilizar las uniones Oracle para crear una consulta que muestre el apellido de los empleados, el nombre de departamento, el salario y el nombre del país de todos los empleados.

```
SELECT em.last_name "last name", dp.department_name "department
name",em.salary, con.country_name "country name"
FROM employees em, departments dp, locations loc, countries con
WHERE em.department_id = dp.department_id
AND
dp.location_id = loc.location_id(+)
AND
loc.country_id = con.country_id(+)
```

20. Vuelva a utilizar la sintaxis de unión de Oracle para modificar la consulta anterior, de modo que incluya también incluye el registro de empleado del empleado sin department_id, 'Grant'.

```
SELECT em.last_name "last name", dp.department_name "department
name",em.salary, con.country_name "country name"
FROM employees em, departments dp, locations loc, countries con
WHERE em.department_id = dp.department_id(+)
AND
dp.location_id = loc.location_id(+)
AND
loc.country_id = con.country_id(+)
```