

Cree estas Tablas

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

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

```
/*cree estas tablas*/  
create table emp  
as select * from hr.employees  
  
create table dept  
as select * from hr.departments
```

Escriba la Consulta

- Problem:

- 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.

- Tablas Usadas:

- user_constraints, user_cons_columns

CONSTRAINT_NAME	CONSTRAINT_TYPE	COLUMN_NAME	POSITION
JHIST_EMP_ID_ST_DATE_PK	P	EMPLOYEE_ID	1
JHIST_EMP_ID_ST_DATE_PK	P	START_DATE	2
JHIST_JOB_FK	D	JOB_ID	1
JHIST_EMP_FK	D	EMPLOYEE_ID	1
JHIST_DEPT_FK	D	DEPARTMENT_ID	1

d

```
/*ejercicio #2*/ /*hecho*/
```

```
select ucons.constraint_name, ucons.constraint_type, col.column_name, col.position
from user_constraints ucons
inner join user_cons_columns col on ucons.constraint_name = col.constraint_name
where col.table_name = 'JOB_HISTORY';
```

Cree la Sentencia

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

Table altered.

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

Table altered.

```
/*ejercicio #3*/  /*hecho */  
  
alter table emp  
add constraint PK_EMP primary key (employee_id);  
  
alter table dept  
add constraint PK_DEPT primary key (department_id)
```

Corrija el Código

- Problem:

- Agregue una restricción ajena entre DEPT y EMP, para que en la tabla EMP solo se puedan introducir departamentos válidos, pero asegúrese de que puede suprimir cualquier fila de la tabla DEPT.

- Sentencia:

```
ALTER TABLE emp  
CREATE CONSTRAINT FOREIGN KEY (dept_id) REFS dept(deptid)  
on del cascade
```

Table altered.

```
/*ejercicio #4*//*hecho*/
```

```
alter table emp  
add constraint emp_dept_department_id_fk foreign key (department_id)  
references dept (department_id) on delete cascade;
```

Cree el Código

- Pruebe la restricción de clave ajena que acaba de crear siguiendo los ejemplos de esta diapositiva.

```
SELECT COUNT(*) AS "Num emps"
FROM emp;
```

Num emps
20

- Examine el número de filas de la tabla EMP. Elimine los detalles del departamento 10 de la tabla dept.

```
DELETE dept
WHERE department_id = 10;
```

1 row(s) deleted.




- Ahora cuente emps de nuevo y compruebe si también hay menos empleados.

```
SELECT COUNT(*) AS "Num emps"
FROM emp;
```

Num emps
19

```
/*ejercicio #5*/
Select count(*) as "num emps" from emp;
Delete dept where department_id = 10;

Select count(*) as "num emps" from emp;
```

Salida de Script x	Resultado de la Consulta x	Resultado de la Co
   SQL Todas las Filas Recuperadas: 1 en 0.002 segundos		
num emps		
1	106	

Escriba la Consulta

- Problem:

- 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.

- Tablas Usadas:

- Employees, Departments

LAST_NAME	SALARY	DEPARTMENT_ID	SALAVG
Hartstein	13000	20	9500
Mourgos	5800	50	3500
Hunold	9000	60	6400
Zlotkey	10500	80	10033
Abel	11000	80	10033
King	24000	90	19333
Higgins	12000	110	10150

ORACLE®

```

/*ejercicio #6*/
WITH salario_medio_por_departamento AS (SELECT NVL(DEPARTMENT_ID, -1) DEPARTMENT_ID, AVG(NVL(SALARY,0)) AVG_SALARIO
FROM hr.EMPLOYEES
GROUP BY NVL(DEPARTMENT_ID, -1))
SELECT EM.LAST_NAME, ROUND(EM.SALARY,2) "SALARY",
CASE
WHEN AVGQUERY.DEPARTMENT_ID = -1 THEN NULL
ELSE AVGQUERY.DEPARTMENT_ID
END "DEPARTMENT_ID", ROUND(AVGQUERY.AVG_SALARIO,2) "SALAVG"
FROM hr.EMPLOYEES EM
INNER JOIN (SELECT * FROM salario_medio_por_departamento) AVGQUERY
ON NVL(EM.DEPARTMENT_ID, -1) = AVGQUERY.DEPARTMENT_ID
WHERE EM.SALARY > AVGQUERY.AVG_SALARIO
ORDER BY 3 ASC

```

Salida de Script x Resultado de la Consulta x Resultado de la Consulta 1 x Resultado de la Consulta 2 x Resultado de la Consulta 3 x Resultado de la C

Todas las Filas Recuperadas: 38 en 0.011 segundos

LAST_NAME	SALARY	DEPARTMENT_ID	SALAVG
1 Hartstein	13000	20	9500
2 Raphaely	11000	30	4150
3 Bell	4000	50	3475,56
4 Dilly	3600	50	3475,56
5 Weiss	8000	50	3475,56
6 Fripp	8200	50	3475,56
7 Kaufling	7900	50	3475,56

Escriba el Código

- Problem:

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

- Tablas Usadas:

- emp, dept

```
SELECT * FROM v2;
```

Nombre del Departamento	Lowest Salary	Highest Salary	Average Salary
Accounting	8300	12000	10150
IT	4200	9000	6400
Executive	17000	24000	19333
Envío	2500	5800	3500
Ventas	8600	11000	10033
Marketing	6000	13000	9500

ORACLE®

```
/*ejercicio #7*/
create or replace view v2
( "nombre departamento", "salario menor", "salario mayor", "salario medio") as
select
  d.department_name,
  to_char(round(min(nvl(e.salary,0)),2),'$999999.99'),
  to_char(round(max(nvl(e.salary,0)),2),'$999999.99'),
  to_char(round(avg(nvl(e.salary,0)),2),'$999999.99')
from hr.departments d
left outer join hr.employees e
on d.department_id = e.department_id
group by (d.department_id, d.department_name);
```

Salida de Script x | Resultado de la Consulta x | Resultado de la Consulta 1 x | Resultado de la Consulta 2 x

Tarea terminada en 0.089 segundos

>Query Run In:Resultado de la Consulta 1
filas eliminado

>Query Run In:Resultado de la Consulta 3
iew V2 creado.

iew V2 creado.

Escriba el Código

- Problem:

- 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.
- Vuelva a ejecutar una sentencia UPDATE en la vista.

DEPT_NAME	MGR_NAME
Executive	S.King
IT	A.Hunold
Envío	K.Mourgos
Ventas	E.Zlotkey
Entorno de	J.Whalen
Marketing	M.Hartstein
Accounting	S.Higgins

- Tablas Usadas:

- Employees, departments

ORACLE
Academy

DPS20L1
Garantía de Resultados de Consultas de
Calidad - Técnicas Avanzadas

Copyright © 2019, Oracle y/o sus filiales. Todos los derechos reservados.

12


```

/*ejercicio #8*/
create or replace view manager_view as
select distinct
  substr(nvl(m.first_name, '_'),1, 1) || '.' || m.last_name apellido,
  d.department_name
from
  hr.employees m inner join hr.employees emp on m.employee_id = emp.manager_id
left outer join hr.departments d on m.department_id = d.department_id;

update dept_managers_view set department_name = (select first_name from employees);

select * from manager_view
  
```

Salida de Script x | Resultado de la Consulta x | Resultado de la Consulta 1 x | Resultado de la Consulta 2 x

 Todas las Filas Recuperadas: 18 en 0.032 segundos

APELLIDO	DEPARTMENT_NAME
1 K.Mourgos	Shipping
2 M.Weiss	Shipping
3 S.King	Executive
4 G.Cambrault	Sales
5 P.Kaufling	Shipping
6 E.Zlotkey	Sales
7 K.Partners	Sales
8 J.Russell	Sales
9 N.Greenberg	Finance

Corrija el Código

- Problem:
 - La siguiente sentencia contiene errores.
 - Corríjalos y ejecute el código para obtener el resultado que se muestra.
- Código:

```
DROP V3 views;
```

View dropped.

The screenshot shows the Oracle SQL Developer interface. The script editor contains the following code:

```
/*ejercicio #9*/  
  
drop view v2;
```

The results window at the bottom shows the output of the script execution:

```
Salida de Script x Resultado de la Consulta x Resultado de la Consulta 1 x Resultado de la Consulta 2 x Resultado de la Consulta 3 x  
Tarea terminada en 0.036 segundos  
left outer join departments d on m.department_id = d.department_id  
Informe de error -  
Error SQL: ORA-00942: table or view does not exist  
00942. 00000 - "table or view does not exist"  
*Cause:  
*Action:  
View V2 borrado.
```

Cree una Secuencia y Corrija el Código

- Problem:
 - Cree una secuencia denominada ct_seq con todos los valores por defecto. Ejecute las sentencias y corrija el error.
 - Corrija la sentencia para que devuelva el número posterior.
- Código:

```
CREATE SEQUENCE ct_seq;
```




Sequence created.

```
SELECT ct_seq.currval
FROM dual;
```

 ORA-08002: sequence CT_SEQ.CURRVAL is not yet defined in this session

```
/**ejercicio #10*/
```

```
create sequence ct_seq ;
select ct_seq.currval from dual;
select ct_seq.nextval from dual;
```

Salida de Script x	Resultado de la Consulta x	Result
		 SQL Todas las Filas Recuperadas: 1 en 0.001
NEXTVAL		
1		24

Corrija el Código

- Problem:
 - Observe la sentencia INSERT y corrija el error.
- Código:

```
INSERT emp
(employee_id,first_name,last_name,email,phone_number,
hire_date,job_id,salary,commission_pct,manager_id,department_id)
VALUES
(currval(ct_seq),'Kaare','Hansen','KHANSEN','44965 832123',sysdate,
'Manager',6500,null,100,10)
```

ORA-00984: column not allowed here

/**ejercicio #11*/

```
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, 'manager', 6500, null, 100, 20);
```

Tarea terminada en 0.004 segundos

Error en la línea de comandos : 197 Columna : 0

Informe de error -
 Error SQL: ORA-00925: missing INTO keyword
 00925. 00000 - "missing INTO keyword"
 *Cause:
 *Action:
 Error que empieza en la línea: 197 del comando -
 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, 'manager', 6500, null, 100,
 Error en la línea de comandos : 198 Columna : 106

Informe de error -
 Error SQL: ORA-00936: missing expression
 00936. 00000 - "missing expression"
 *Cause:
 *Action:
 1 fila insertadas.

Corrija el Código

- Problem:

- Corrija el error en la sentencia SQL para crear el índice como se muestra en la captura de pantalla.

- Código:

```
CREATE INX emp indx FOR TABLE emp(employee_id DESC,
UPPR(SUBST(firstname,1,1) || " " || lastname)
```

TABLE_NAME	INDEX_NAME	INDEX_TYPE	COLUMN_EXPRESSION	COLUMN_POSITION
EMP	EMP_INDX	FUNCTION-BASED NORMAL	"EMPLOYEE_ID"	1
EMP	EMP_INDX	FUNCTION-BASED NORMAL	UPPER(SUBSTR("FIRST_NAME",1,1) ' ' "LAST_NAME")	2

ORACLE®

```
/**ejercicio #12*/
create inx emp indx
for table emp (employee_id desc, uppr(subst(firstname,1,1) || " " || last_name));
create index emp_index on emp (employee_id desc, upper(substr(first_name,1,1) || ' ' || last_name));
```

Salida de Script x Resultado de la Consulta x Resultado de la Consulta 1 x Resultado de la Consulta 2 x Resultado de la Consulta 3

Tarea terminada en 0.082 segundos

Error SQL: ORA-00901: invalid CREATE command
 901. 00000 - "invalid CREATE command"
 Cause:
 Action:
 Index EMP_INDEX creado.

Escriba el Código

- Problem:
 - Escriba la sentencia SQL para mostrar todas las tablas de usuario que contienen el nombre PRIV.
- Tablas Usadas:
 - dictionary

TABLE_NAME	COMMENTS
USER_AQ_AGENT_PRIVS	-
USER_COL_PRIVS	Concesiones en las columnas cuyo propietario, otorgante o usuario con privilegios es el usuario
USER_COL_PRIVS_MADE	Todos los privilegios otorgados sobre las columnas de objetos propiedad del usuario
USER_COL_PRIVS_RECD	Concesiones en las columnas cuyo usuario con privilegios es el usuario
USER_GOLDENGATE_PRIVILEGES	Detalles de privilegios de GoldenGate
USER_NETWORK_ACL_PRIVILEGES	Privilegios de usuario para acceder a los hosts de red mediante los paquetes de utilidades de red PL/SQL
USER_REPGROUP_PRIVILEGES	Información sobre los usuarios que se han registrado para los privilegios de grupo de objetos
USER_ROLE_PRIVS	Roles otorgados al usuario actual
USER_RSRC_CONSUMER_GROUP_PRIVS	Privilegios de cambio para grupos de consumidores del usuario
USER_RSRC_MANAGER_SYSTEM_PRIVS	Privilegios del sistema para el gestor de recursos para del usuario
...	...

ORACLE

```

/**ejercicio #13*/

select * from dictionary
where table_name like upper('%priv%');

/**ejercicio #14*/

```

Salida de Script x | Resultado de la Consulta x | Resultado de la Consulta 1 x | Resultado de la Consulta 2 x | Resultado de la Consulta 3 x | Resultado

Todas las Filas Recuperadas: 43 en 0.013 segundos

TABLE_NAME	COMMENTS
1 DBA_AQ_AGENT_PRIVS	(null)
2 DBA_COL_PRIVS	All grants on columns in the database
3 DBA_GOLDENGATE_PRIVILEGES	Details about goldengate privileges
4 DBA_NETWORK_ACL_PRIVILEGES	Privileges defined in network access control lists
5 DBA_PRIV_AUDIT_OPTS	Describes current system privileges being audited across the system and by user
6 DBA_REPGROUP_PRIVILEGES	Information about users who are registered for object group privileges
7 DBA_ROLE_PRIVS	Roles granted to users and roles
8 DBA_RSRC_CONSUMER_GROUP_PRIVS	Switch privileges for consumer groups

Línea 337 Columna 1

Corrija el Código

- Problem:
 - Conceda acceso de selección a público en la tabla EMP y verifique que se ha otorgado mediante la ejecución esta consulta. La consulta contiene errores que debe corregir antes de poder ejecutar la sentencia SELECT.
- Código:

```
GRANT SELECT ON emp TO PUBLIC
```

Statement processed.

```
SELECT *
FROM usr_tab_privs
WHERE tablename = "emp"
```

GRANTEE	OWNER	TABLE_NAME	GRANTOR	PRIVILEGE	GRANTABLE	HIERARCHY
PUBLIC	US_A009EMEA815_PLSQL_T01	EMP	US_A009EMEA815_PLSQL_T01	SELECT	NO	NO

```
/**ejercicio #14*/
```

```
grant select on emp to public;
select * from user_tab_privs where table_name = 'emp';
```

Salida de Script x Resultado de la Consulta x Resultado de la Consulta 1 x Resultado de la Consulta 2 x Resultado de la Consulta 3 x

SQL | Todas las Filas Recuperadas: 0 en 0.027 segundos

GRANTEE	OWNER	TABLE_N...	GRANTOR	PRIVILEGE	GRANTABLE	HIERARCHY
---------	-------	------------	---------	-----------	-----------	-----------

Escriba el Código

- Problem:
 - Mediante las uniones propiedad de Oracle, construya una instrucción que devuelva todos los employee_id unidos a todos los department_names.
- Tablas Usadas:
 - Employees, departments

104	Contracting
107	Contracting
124	Contracting
141	Contracting
142	Contracting
143	Contracting
144	Contracting
149	Contracting
174	Contracting
176	Contracting
178	Contracting
200	Contracting
201	Contracting
202	Contracting
205	Contracting
206	Contracting

160 rows returned in 0.00 seconds

ORACLE®

```
/**ejercicio #15*/
```

```
select e.employee_id, d.department_name from hr.employees e, hr.departments d;
```

Salida de Script × | Resultado de la Consulta × | Resultado de la Consulta 1 × | Resultado de la Consulta 2 × | Resu

SQL | Se han recuperado 50 filas en 0.003 segundos

	EMPLOYEE_ID	DEPARTMENT_NAME
1	100	Administration
2	101	Administration
3	102	Administration
4	103	Administration
5	104	Administration
6	105	Administration
7	106	Administration
8	107	Administration
9	108	Administration
10	109	Administration

Escriba el Código

- Problem:

- 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.

- Tablas Usadas:

- Employees, departments

EMPLOYEE_ID	DEPARTMENT_NAME
200	Entorno de
201	Marketing
202	Marketing
124	Envío
144	Envío
143	Envío
142	Envío
141	Envío
107	IT
104	IT
103	IT
174	Ventas
149	Ventas
176	Ventas
102	Executive
100	Executive
101	Executive
205	Accounting
206	Accounting

ORACLE®

```
/**ejercicio #16*/
```

```
select e.employee_id, d.department_name from hr.employees e, hr.departments d
where e.department_id = d.department_id;
```

lida de Script x | Resultado de la Consulta x | Resultado de la Consulta 1 x | Resultado de la Consulta 2

SQL | Se han recuperado 50 filas en 0.336 segundos

	EMPLOYEE_ID	DEPARTMENT_NAME
1	200	Administration
2	201	Marketing
3	202	Marketing
4	114	Purchasing
5	115	Purchasing
6	116	Purchasing
7	117	Purchasing
8	118	Purchasing
9	119	Purchasing
10	203	Human Resources

Escriba el Código

- Problem:

- 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.

- Tablas Usadas:

- Employees, departments, locations y countries

LAST_NAME	DEPARTMENT_NAME	SALARY	COUNTRY_NAME
King	Executive	24000	United States of America
Kochhar	Executive	17000	United States of America
De Haan	Executive	17000	United States of America
Whalen	Administration	4400	United States of America
Higgins	Accounting	12000	United States of America
Gietz	Accounting	8300	United States of America
Zlotkey	Sales	10500	United Kingdom
Abel	Sales	11000	United Kingdom
Taylor	Sales	8600	United Kingdom
Mourgos	Shipping	5800	United States of America
Rajs	Shipping	3500	United States of America
Davies	Shipping	3100	United States of America
Matos	Shipping	2600	United States of America
Vargas	Shipping	2500	United States of America
Hunold	IT	9000	United States of America
Ernst	IT	6000	United States of America
Lorentz	IT	4200	United States of America
Hartstein	Marketing	13000	Canada
Fay	Marketing	6000	Canada

ORACLE®

```
/**ejercicio #17*/
```

```
select e.last_name "apellido", d.department_name "departamento", e.salary "salario", c.country_name "pais"
from hr.employees e, hr.departments d, hr.locations l, hr.countries c
where e.department_id = d.department_id and d.location_id = l.location_id and l.country_id = c.country_id;
```

Salida de Script x | Resultado de la Consulta x | Resultado de la Consulta 1 x | Resultado de la Consulta 2 x | Resultado de la Consulta 3 x | Resultado de...

apellido	departamento	salario	pais
1 King	Executive	24000	United States of America
2 Kochhar	Executive	17000	United States of America
3 De Haan	Executive	17000	United States of America
4 Hunold	IT	9000	United States of America
5 Ernst	IT	6000	United States of America
6 Austin	IT	4800	United States of America
7 Pataballa	IT	4800	United States of America
8 Lorentz	IT	4200	United States of America
9 Greenberg	Finance	12008	United States of America
10 Faviat	Finance	9000	United States of America

Escriba el Código

- Problem:

- 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'.

- Tablas Usadas:

- Employees, departments, locations y countries

LAST_NAME	DEPARTMENT_NAME	SALARY	COUNTRY_NAME
Hartstein	Marketing	13000	Canada
Fay	Marketing	6000	Canada
Zlotkey	Sales	10500	United Kingdom
Abel	Sales	11000	United Kingdom
Taylor	Sales	8600	United Kingdom
Hunold	IT	9000	United States of America
Ernst	IT	6000	United States of America
Lorentz	IT	4200	United States of America
Mourgos	Shipping	5800	United States of America
Rajs	Shipping	3500	United States of America
Davies	Shipping	3100	United States of America
Matos	Shipping	2600	United States of America
Vargas	Shipping	2500	United States of America
Higgins	Accounting	12000	United States of America
Gietz	Accounting	8300	United States of America
King	Executive	24000	United States of America
Kochhar	Executive	17000	United States of America
De Haan	Executive	17000	United States of America
Whalen	Administration	4400	United States of America
Grant	-	7000	-

```
/**ejercicio #18*/
```

```
select e.last_name "apellido", d.department_name "departamento", e.salary "salario", c.country_name "pais"
from hr.employees e, hr.departments d, hr.locations l, hr.countries c
where e.department_id = d.department_id(+) and d.location_id = l.location_id(+) and l.country_id = c.country_id(+);
```

Salida de Script x	Resultado de la Consulta x	Resultado de la Consulta 1 x	Resultado de la Consulta 2 x	Resultado de la Consulta 3 x	Resultado d
<div> </div> Todas las Filas Recuperadas: 107 en 0.03 segundos					
apellido	departamento	salario	pais		
98 Smith	Sales	7400	United Kingdom		
99 Bates	Sales	7300	United Kingdom		
100 Kumar	Sales	6100	United Kingdom		
101 Abel	Sales	11000	United Kingdom		
102 Hutton	Sales	8800	United Kingdom		
103 Taylor	Sales	8600	United Kingdom		
104 Livingston	Sales	8400	United Kingdom		
105 Johnson	Sales	6200	United Kingdom		
106 Baer	Public Re...	10000	Germany		
107 Grant	(null)	7000	(null)		