

# Consultas-Practicas-SQL.pdf



**Zukii**



**Fundamentos de Bases de Datos**



**2º Grado en Ingeniería Informática**

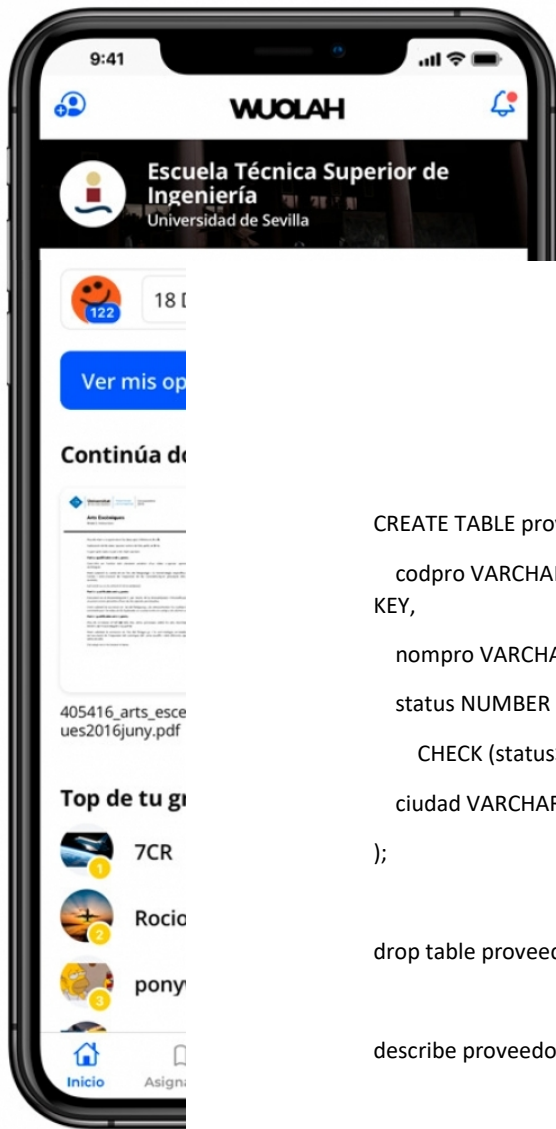


**Escuela Técnica Superior de Ingenierías Informática y de Telecomunicación**  
**Universidad de Granada**



**Descarga la APP de Wuolah.**  
Ya disponible para el móvil y la tablet.





# Descarga la APP de Wuolah.

Ya disponible para el móvil y la tablet.



```
CREATE TABLE proveedor(
```

```
    codpro VARCHAR2(3) CONSTRAINT cod_pro NOT NULL CONSTRAINT cod_prim PRIMARY KEY,
```

```
    nompro VARCHAR2(30) CONSTRAINT nompro_no_nulo NOT NULL,
```

```
    status NUMBER CONSTRAINT status_entre_1_y_10
```

```
        CHECK (status>=1 and status<=10),
```

```
    ciudad VARCHAR2(15)
```

```
);
```

```
drop table proveedor
```

```
describe proveedor
```

```
CREATE TABLE pieza(
```

```
    codpie VARCHAR2(3) CONSTRAINT codpie_clave_primaria PRIMARY KEY,
```

```
    nompie VARCHAR2(10) CONSTRAINT npmpie_no_nulo NOT NULL,
```

```
    color VARCHAR2(10),
```

```
    peso NUMBER(5,2) CONSTRAINT peso_entre_0_y_100 CHECK (peso>0 and peso<=100),
```

```
    ciudad VARCHAR2(15)
```

```
);
```

```
describe pieza
```

```
CREATE TABLE proyecto(
```

```
    codpj VARCHAR2(3) CONSTRAINT codpj_clave_primaria PRIMARY KEY,
```

```
    nompj VARCHAR2(20) CONSTRAINT nompj_no_nulo NOT NULL,
```

```
    ciudad VARCHAR2(15)
```

```
);
```

```
describe proyecto
```

```
CREATE TABLE ventas(  
    codpro CONSTRAINT codpro_ce_proveedor REFERENCES proveedor(codpro),  
    codpie CONSTRAINT codpie_ce_pieza REFERENCES pieza(codpie),  
    codpj CONSTRAINT copj_ce_royecto REFERENCES proyecto(codpj),  
    cantidad NUMBER(4),  
    CONSTRAINT clave_primaria PRIMARY KEY (codpro, codpie, codpj)  
);
```

describe ventas

```
ALTER TABLE ventas ADD fecha date
```

describe ventas

```
select codpro from ventas
```

```
select table_name from user_tables
```

describe proveedor

```
INSERT INTO proveedor values ('S1', 'Jose Fernandez', 2, 'Madrid')
```

```
INSERT INTO proveedor values ('S2', 'Manuel Vidal', 1, 'Londres')
```

```
INSERT INTO proveedor values ('S3', 'Luisa Gomez', 3, 'Lisboa')
```

```
INSERT INTO proveedor values ('S4', 'Pedro Sanchez', 4, 'Paris')
```

```
INSERT INTO proveedor values ('S5', 'Maria Reyes', 5, 'Roma')
```

```
select * from proveedor
```

```
INSERT INTO pieza values ('P1', 'Tuerca', 'Gris', 2.5, 'Madrid');
```

```
INSERT INTO pieza values ('P2', 'Tornillo', 'Rojo', 1.25, 'Paris');
```

```
INSERT INTO pieza values ('P3', 'Arandela', 'Blanco', 3, 'Londres');
```

```
INSERT INTO pieza values ('P4', 'Clavo', 'Gris', 5.5, 'Lisboa');
```

```
INSERT INTO pieza values ('P5', 'Alcayata', 'Blanco', 10, 'Roma');
```

```
select * from pieza
```



**KEEP  
CALM  
AND  
ESTUDIA  
UN POQUITO**

describe proyecto

INSERT INTO proyecto values ('J1', 'Proyecto 1', 'Londres');

INSERT INTO proyecto values ('J2', 'Proyecto 2', 'Londres');

INSERT INTO proyecto values ('J3', 'Proyecto 3', 'Paris');

INSERT INTO proyecto values ('J4', 'Proyecto 4', 'Roma');

select \* from proyecto

select \* from opc.ventas

insert into ventas select \* from opc.ventas

select \* from ventas

/\*sesion 2 \*/

select \* from pieza where (color='Rojo' OR color='Gris') AND (ciudad='Madrid')

select \* from pieza where upper(color) in ('GRIS', 'ROJO') AND (ciudad='Madrid') /\*para que no importe las mayusculas\*/

SELECT \* from ventas where (cantidad BETWEEN 200 and 300)

/\*

select \* from ventas

update ventas set fecha = fecha + 365\*4 where fecha<'01/01,/17'>

rollback \*/

-- PIEZAS VENDIDAS POR S1 ORDENADAS Y SIN REPETIDOS

select DISTINCT codpie from ventas where (codpro = 'S1') ORDER BY codpie desc

--ENCONTRAR LOS PROVEEDORES QUE HAN VENDIDO P1 O P5

select codpro from ventas where (codpie = 'P1' OR codpie='P5')

--PROYECTOS Y DINERO GASTADO EN COMPRA DE LA PIEZA P4 SABIENDO QUE CADA UD VALE 2.3€

```
select codpj, cantidad, 2.30*cantidad from ventas where (codpie='P4')
```

-- SES 3

```
select * from proveedor, ventas where proveedor.codpro=ventas.codpro
```

-- pr es alias de proveedor y v de ventas

```
select * from proveedor pr, ventas v where pr.codpro=v.codpro
```

--si hay ambigüedad, ponemos delante nombre de tabla

```
select * from ventas v, pieza p where v.codpie=p.codpie and color='Blanco' and ciudad='Londres' and fecha >='01/01/21'
```

--ventas en las que pr, p y proyecto sean de la misma ciudad

```
select v.codpro, v.codpie, v.codpj, pi.ciudad
```

```
from proveedor pr, pieza pi, ventas v, proyecto pj
```

```
where pr.ciudad=pi.ciudad and pr.codpro=v.codpro and v.codpie=pi.codpie
```

```
and v.codpj=pj.codpj and pi.ciudad=pj.ciudad
```

```
order by v.codpro, v.codpie
```

--parejas de proveedores que no vivan en la misma ciudad

```
select p1.codpro, p1.nompro , p2.codpro, p2.nompro from proveedor p1, proveedor p2
```

```
where p1.codpro<p2.codpro and p1.ciudad<>p2.ciudad --con lo primero hace que solo se relacionen 1 veces entre ellos
```

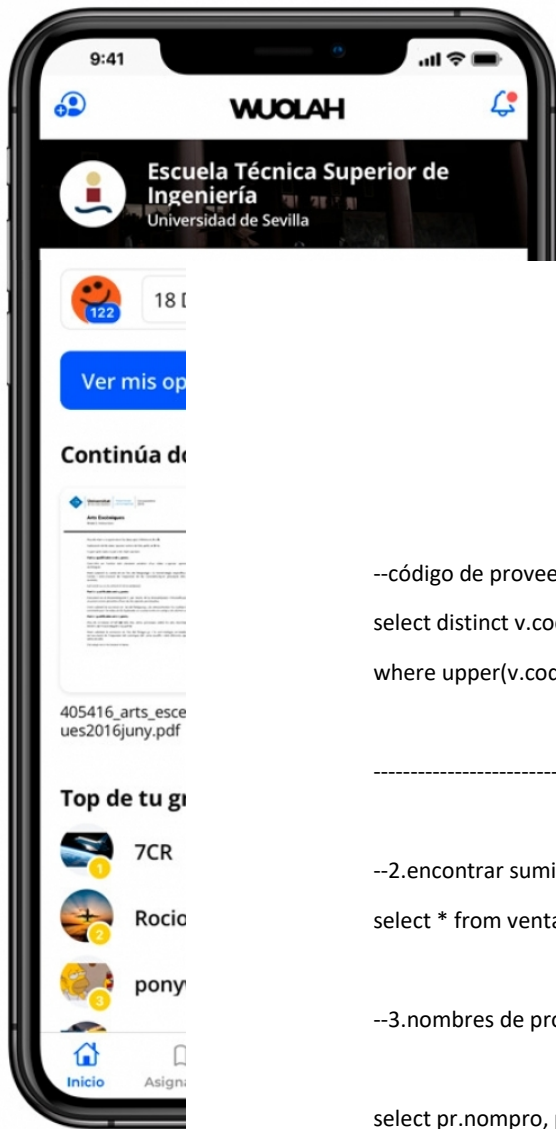
--para ventas de mas de 800 cant mostrar nombre de proveedor y cantidad

```
select distinct pr.nompro, v.cantidad from proveedor pr, ventas v where v.cantidad>800
```

--mostrar piezas (saber su nombre) vendidas por proveedores de madrid

```
select distinct pi.* from pieza pi, proveedor pr, ventas v
```

```
where pr.ciudad='Madrid' and v.codpie=pi.codpie and v.codpro=pr.codpro
```



# Descarga la APP de Wuolah.

Ya disponible para el móvil y la tablet.



--código de proveedores que venden alguna pieza a J4

```
select distinct v.codpro from ventas v
where upper(v.codpj)='J4'
```

--2.encontrar suministros (ventas) cuya cantidad supere las 100

```
select * from ventas v where v.cantidad>100
```

--3.nombres de proveedores, proyectos y piezas que son de la misma ciudad

```
select pr.nompro, pi.nompie, pj.nompj, pi.ciudad
from proveedor pr, pieza pi, ventas v, proyecto pj
where pr.ciudad=pi.ciudad and pr.codpro=v.codpro and v.codpie=pi.codpie
and v.codpj=pj.codpj and pi.ciudad=pj.ciudad
```

--4.nombres de piezas suministradas por Londres

```
select pi.nompie from pieza pi, proveedor pr, ventas v
where pr.ciudad='Londres' and v.codpie=pi.codpie and v.codpro=pr.codpro
```

--5.Parejas de ciudad que la primera sea de un proveedor y la segunda de un proy

-- entre los cuales haya un suministro

```
select distinct pr.ciudad, pj.ciudad from proveedor pr, proyecto pj, ventas v
where v.codpro=pr.codpro and v.codpj=pj.codpj
```

--6.código de proyectos que tienen al menos un proveedor que no se encuentre en

-- su ciudad

```
select distinct pj.codpj from proyecto pj, proveedor pr
where pj.ciudad<>pr.ciudad
```

--7.encontrar códigos de los proyectos que usan una pieza que fue suministrada

-- por S1

```
select distinct pj.codpj from proyecto pj, proveedor pr, ventas v
where v.codpro='S1'
```

--8.encontrar la cantidad más pequeña enviada en algún suministro - Sesión 3 (all y any)

```
select cantidad from ventas where cantidad <= ALL (select cantidad from ventas)
```

--SESION 3

```
(select codpie from pieza where color='Blanco') UNION (select codpie from ventas where
codpro='S2') --se podria hacer con PC
```

--Union all no, Union eliminara valores repetidos

```
(select codpie from pieza where color='Blanco') INTERSECT (select codpie from ventas where
codpro='S2') --se podria hacer con PC
```

```
(select codpie from pieza where color='Blanco') MINUS (select codpie from ventas where
codpro='S2')-- los que cumplen - los que no
```

```
--select dni from alumnos where provincia=(select provincia from alumnos where
nom_alum='JUAN LOPEZ')
```

```
--select dni from matriculas where cod_asig in (select cod_asig from asignaturas where
caracter='op')
```

-- not in

```
--select dni from alumnos where intersection select dni from
```

--in es igual a alguno y not in no aparece

-- EJERCICIOS POST SESION 3

--piezas vendidas por proveedores de madrid con IN

```
select codpie from pieza where codpie in (select codpie from ventas where codpro in (select
codpro from proveedor where ciudad='Madrid')) order by codpie
```



--proyectos en ciudad donde se fabrica alguna pieza

select codpj from proyecto where ciudad in (select ciudad from pieza)

--código de proyectos que no utilizan pieza roja que este suministrada por londres

--(select codpj from proyecto) MINUS (select codp...

--(select codpie from pieza where color='Rojo') INTERSECT (select codpie from ventas...

-- se podría hacer con conjuntos pero es mas largo

select codpj from proyecto where codpj NOT IN (select codpj from ventas where codpie in (select codpie from pieza where color = 'Rojo') and codpro in (select codpro from proveedor where ciudad='Londres'))

--códigos de proveedores que han vendidos piezas P1 y P4

select codpro from ventas where codpie in (select codpie from pieza where codpie='P1' or codpie='P4')

select codpro from ventas where codpie in ((select codpie from pieza where codpie='P1') union (select codpie from pieza where codpie='P4'))

select \* from ventas order by codpie

--código de piezas que solo han sido vendidas por s4

(select codpie from ventas where codpro='S4') MINUS (select codpie from ventas where codpro<>'S4')

--nombre y código de la pieza que pesa más

select codpie, nompie, peso from pieza where peso <= ALL (select peso from pieza)

----nombres de todos que han vendidos piezas P1 y P4

select distinct nompro, nompie, nompj from proveedor, pieza, proyecto where codpro in (select codpro from ventas where codpie in (select codpie from pieza where codpie='P1' or codpie='P4'))

--Sesion 5

--número de ventas con cantidad >1000

select count(cantidad) from ventas where cantidad>1000

--mostrar peso maximo

select max(peso) from pieza

--mostrar código pieza de mas pesada

select codpie from pieza where peso=(select max(peso) from pieza)

--proveedores que han hecho mas de 3 envios diferentes

select distinct codpro from ventas v1 where (select count(\*) from ventas v2 where v2.codpro=v1.codpro) >3

select count(\*) from ventas where codpro='S1' --filosofia

--nombres de proveedores que hayan vendido más de 1000 unidades

select nompro from proveedor where codpro in (select distinct v1.codpro from ventas v1 where(select sum(cantidad) from ventas v2 where v2.codpro=v1.codpro) > 1000)

--Sesión 6 - Group By & Having

--Nombres de proveedores tales que el total de ventas superen las 100

select nompro from proveedor where codpro IN (select codpro from (select codpro, sum(cantidad) from ventas group by codpro having sum(cantidad)>100 order by codpro))

--Cantidad total de cada pieza enviada a cada proyecto

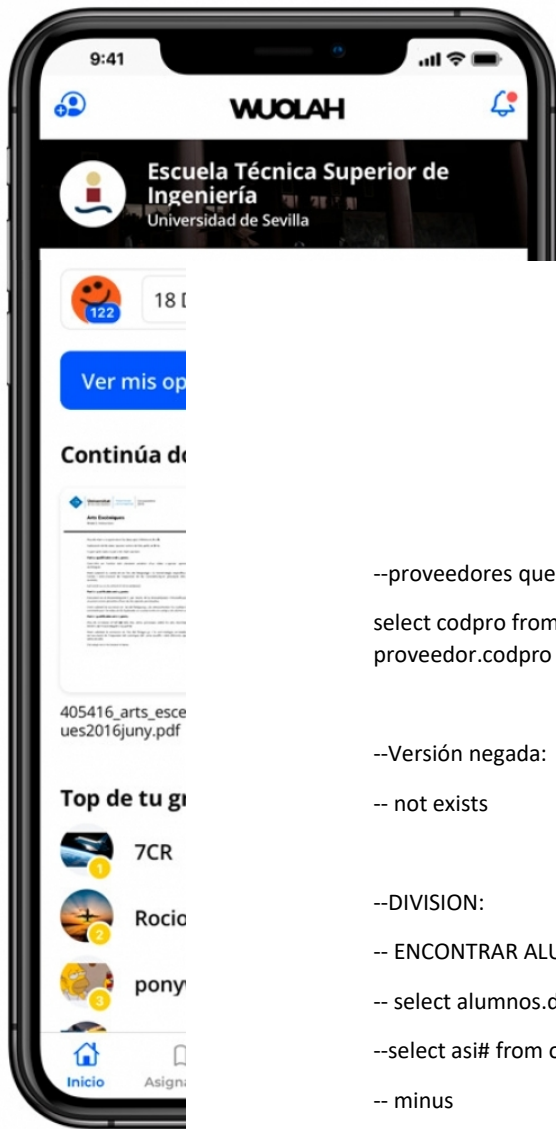
select codpie, codpro, sum(cantidad) from ventas group by codpie, codpro order by codpro, codpie

--Cantidad media de ventas de la pieza P1 realizadas por cada proveedor

select codpro, avg(cantidad) from ventas where codpie='P1' group by codpro order by codpro

--Where Exist, devuelve true si tabla tiene algo

--Se puede usar como In o reunion natural



# Descarga la APP de Wuolah.

Ya disponible para el móvil y la tablet.



--proveedores que suministran pieza p1

```
select codpro from proveedor where exists (select * from ventas where ventas.codpro =
proveedor.codpro and ventas.codpie='P1')
```

--Versión negada: Operador como la diferencia (ej proveedores que NO suministran pieza p1)

-- not exists

--DIVISION:

-- ENCONTRAR ALUMNOS MATRICULADOS DE TODAS LAS ASIGNATURAS DE SEGUNDO

```
-- select alumnos.dni from alumnos where not exist(
```

```
--select asi# from opc.asigna where curso=2
```

-- minus

```
--select codasi# from matricula where matricula.DNI = alumnos.DNI)
```

--Alumnos matriculados de todo segundo en el mismom curso

```
-- select m1.DNI, m1.CursoAcademico from matricula m1 where not exist(
```

```
--select asi# from opc.asigna where curso=2
```

-- minus

```
--select codasi# from matricula m2 where m1.DNI = m2.DNI and m2.cursoAcademico =
m1.cursoAcademico )
```

-- Sesión 9 - Dar nombre a consulta

```
create view media_18 as select codpie, avg(cantidad) media from ventas
```

```
where to_char(fecha,'yyyy')='2018' group by codpie
```

```
select * from media_18
```

-- Las complejas, solo para consultar

-- Problemas: mas de una tabla (de origen) o me falten columna -> Particionar horizontalmente

--Catálogo -> BD que gestiona BD

```
select table_name, tablespace_name, Pct_free from user_tables
```

```
select table_name, column_name, data_type, data_length, nullable from user_tab_columns
```

```
describe user_constraints
```

```
select constraint_name, constraint_type, table_name, search_condition from user_constraints
```

```
select constraint_name, table_name, column_name from user_cons_columns
```

```
-- si en ver de user_ ponemos all_, veremos todo de los usuarios que tengamos permisos para verlo
```

```
select table_name, owner from all_tables
```

```
select * from user_indexes
```

```
select * from user_views
```