File   Edit   Selection   View   Go   Run   Terminal   Help

Project2_scripts.sql 2   Results: Project2_scripts.sql ×

```
SQL> -- 1) Extract the Month, Day & Year I three different columns in Calendar Table. If Table not created please create the table based on the file received.
SQL> select extract (month from dates) as Month,
   2         extract (day from dates) as Day,
   3         extract (year from dates) as Year
   4  from calendars2;
```

Select all rows ▾ | Save as: CSV ▾ | Maximum of 500 rows displayed (modify via MaxRows setting)

MONTH	DAY	YEAR
7	12	2016
7	13	2016
7	14	2016
7	15	2016
7	16	2016
7	17	2016
7	18	2016
7	19	2016
7	20	2016
7	21	2016

Page 1 of 50 |< ◄ ► >| (1-10 of 500 rows)

```
SQL> -- 3) Write a query to find out the number of customer who are married.
SQL> select distinct sum(case when MARITALSTATUS='Y' then 1 else 0 end)
   2         over() as "COUNT of Married customers"
   3  from customers2;
```

Select all rows ▾ | Save as: CSV ▾

COUNT of Married customers
8331

9817

Page 1 of 1 |< ◄ ► >| (1-1 of 1 rows)

File   Edit   Selection   View   Go   Run   Terminal   Help

Project2_scripts.sql 2   Results: Project2_scripts.sql ×

```
SQL> -- 5) Write a query to find out how many customers have 0 kids.
SQL> select distinct sum( case when TotalChildren=0 then 1 else 0 end)
   2         over() as "Cutomer COunt with no child"
   3  from customers2;
```

Select all rows ▾ | Save as: CSV ▾

Cutomer COunt with no child
5080

Page 1 of 1 |< ◄ ► >| (1-1 of 1 rows)

```
SQL> -- 7) Give me a count of customers who have their own property.
SQL> select distinct sum( case when HOMEOWNER='Y' then 1 else 0 end)
   2         over() as "Count of customer with own property"
   3  from customers2;
```

Select all rows ▾ | Save as: CSV ▾

Count of customer with own property
12260

Page 1 of 1 |< ◄ ► >| (1-1 of 1 rows)

```
SQL> -- 8) Write a SQL Query to find out the Customer Last Name starts with 'RA' & FIRST Name ending with 'DA'and ensure there is no duplicate records seen when the output is displayed.
SQL> select * from customers2
   2  where lastname like 'RA%' and firstname like '%DA';
```

Select all rows ▾ | Save as: CSV ▾

CUSTOMERKEY	PREFIX	FIRSTNAME	LASTNAME	BIRTHDATE	MARITALSTATUS	GENDER	EMAILADDRESS	ANNUALINCOME	TOTALCHILDREN	EDUCATIONLEVEL	OCCUPATION	HOMEOWNER	FULL_NAME	SALARY
13740	MRS.	JADA	RAMIREZ	07-OCT-57	M	F	jada3@adventure-works.com	$40,000	1	Partial College	Clerical	Y	MRS. JADA RAMIREZ	40000
15017	MS.	AMANDA	RAMIREZ	22-OCT-80	S	F	amanda16@adventure-works.com	$20,000	0	Partial College	Skilled Manual	N	MS. AMANDA RAMIREZ	20000
18019	MRS.	YOLANDA	RAJI	23-DEC-63	M	F	yolanda20@adventure-works.com	$40,000	1	Bachelors	Skilled Manual	Y	MRS. YOLANDA RAJI	40000
23598	MRS.	LINDA	RAMOS	21-NOV-57	S	F	linda32@adventure-works.com	$150,000	2	Partial College	Professional	N	MRS. LINDA RAMOS	150000
20852	MS.	YOLANDA	RAJE	06-JUL-70	M	F	yolanda13@adventure-works.com	$30,000	0	Bachelors	Clerical	Y	MS. YOLANDA RAJE	30000

File   Edit   Selection   View   Go   Run   Terminal   Help

Project2_scripts.sql 2   Results: Project2_scripts.sql ×

23598	MRS.	LINDA	RAMOS	21-NOV-57	S	F	linda32@adventure-works.com	$150,000	2	Partial College	Professional	N	MRS. LINDA RAMOS	150000
20852	MS.	YOLANDA	RAJE	06-JUL-70	M	F	yolanda13@adventure-works.com	$30,000	0	Bachelors	Clerical	Y	MS. YOLANDA RAJE	30000
27047	MRS.	YOLANDA	RAI	13-OCT-58	M	F	yolanda16@adventure-works.com	$40,000	1	Partial College	Clerical	Y	MRS. YOLANDA RAI	40000
20794	MRS.	BRENDA	RANA	08-FEB-62	M	F	brenda15@adventure-works.com	$10,000	1	Graduate Degree	Manual	Y	MRS. BRENDA RANA	10000
29220	MRS.	MELINDA	RAMOS	23-MAR-64	S	F	melinda12@adventure-works.com	$130,000	2	Graduate Degree	Management	N	MRS. MELINDA RAMOS	130000

Page 1 of 1 |< ◄ ► >| (1-8 of 8 rows)

```
SQL> -- 9) Write a query to display the sales for the order date 03/21/2017 for product key 540.
SQL> select * from SALES_2017
   2  where PRODUCTKEY=540 and ORDERDATE= date '2017-03-21';
```

Select all rows ▾ | Save as: CSV ▾

ORDERDATE	STOCKDATE	ORDERNUMBER	PRODUCTKEY	CUSTOMERKEY	TERRITORYKEY	ORDERLINEITEM	ORDERQUANTITY
21-MAR-17	29-FEB-04	SO66588	540	13760	6	1	1
21-MAR-17	09-FEB-04	SO66586	540	11711	6	2	1
21-MAR-17	01-MAR-04	SO66585	540	24490	1	1	1
21-MAR-17	29-JAN-04	SO66587	540	13988	6	1	1
21-MAR-17	16-JAN-04	SO66589	540	24999	4	1	1

Page 1 of 1 |< ◄ ► >| (1-5 of 5 rows)

```
SQL> -- 10) Write a SQL Query to increase the cost of products by 18% and round the data to the nearest number.
SQL> select PRODUCTCOST,floor(PRODUCTCOST*1.18) as FInalPRODUCTCOST
   2  from products2;
```

Select all rows ▾ | Save as: CSV ▾

PRODUCTCOST	FINALPRODUCTCOST
706.811	834
623.8403	736
617.0281	728
699.0928	824
699.0928	824
617.0281	728

Go Live

ENG 21:08 20-11-2022

Project2_scripts.sql 2   Results: Project2_scripts.sql

617.0281	728
699.0928	824
699.0928	824
617.0281	728
617.0281	728
623.8403	736
2171.2942	2562
2171.2942	2562

Page 1 of 30 |< ◀ ▶ >| (1-10 of 293 rows)

```
SQL> -- 11) Adventure work Head of sales would like to find out the cost difference between productcost and product price.
SQL> select PRODUCTPRICE-PRODUCTCOST
  2  from products2;
```

Select all rows   Save as: CSV

PRODUCTPRICE-PRODUCTCOST
657.689
580.4845
574.1458
650.5072
650.5072
574.1458
574.1458
580.4845
1406.9758
1406.9758

Page 1 of 30 |< ◀ ▶ >| (1-10 of 293 rows)

Project2_scripts.sql 2   Results: Project2_scripts.sql

```
SQL> -- 12) Write a SQL Query to find out, which products were not returned (Use tables Product & Returns) solve the query without 'not in' function.
SQL> select p.productkey,p.productname from products2  p
  2  where  not exists (
  3     select 'pkey' from returns2 r
  4     where p.productkey=r.PRODUCTKEY
  5  )order by 1;
```

Select all rows   Save as: CSV

PRODUCTKEY	PRODUCTNAME
218	Mountain Bike Socks, M
219	Mountain Bike Socks, L
238	HL Road Frame - Red, 62
241	HL Road Frame - Red, 44
244	HL Road Frame - Red, 48
247	HL Road Frame - Red, 52
250	HL Road Frame - Red, 56
253	LL Road Frame - Black, 58
256	LL Road Frame - Black, 60
259	LL Road Frame - Black, 62

Page 1 of 17 |< ◀ ▶ >| (1-10 of 169 rows)

```
SQL> -- 13) Write a query to find out which customer has placed most number of sales.
SQL> select CUSTOMERKEY,count(*) as OrderCount
  2  from SALES_2017
  3  group by CUSTOMERKEY
  4  order by count(*) desc;
```

Select all rows   Save as: CSV   Maximum of 500 rows displayed (modify via MaxRows setting)

CUSTOMERKEY	ORDERCOUNT
11300	45
11185	45
11262	40
11276	36
11223	33

Project2_scripts.sql 2   Results: Project2_scripts.sql

11091	30
11277	29
11711	28
11566	25

Page 1 of 50 |< ◀ ▶ >| (1-10 of 500 rows)

```
SQL> -- 14) Write a SQL Query to find out the products returned for Region Germany.
SQL> TERRITORIES2 RETURNS2
ERROR at line 53:
unknown command "TERRITORI..." - rest of line ignored.
SQL> with return_data as (
  2  select * from returns2 r
  3  where exists (
  4     select 'row' from TERRITORIES2 t
  5     where t.COUNTRY='Germany' and  r.TERRITORYKEY=t.SALESTERRITORYKEY
  6  ))select p.productkey,r.TERRITORYKEY,p.productname from products2 p
  7     join return_data r on r.productkey=p.productkey;
```

Select all rows   Save as: CSV

PRODUCTKEY	TERRITORYKEY	PRODUCTNAME
311	8	Road-150 Red, 44
314	8	Road-150 Red, 56
322	8	Road-650 Red, 60
328	8	Road-650 Red, 48
328	8	Road-650 Red, 48
330	8	Road-650 Red, 52
334	8	Road-650 Black, 60
346	8	Mountain-100 Silver, 44
350	8	Mountain-100 Black, 44
352	8	Mountain-200 Silver, 38

Page 1 of 17 |< ◀ ▶ >| (1-10 of 161 rows)

File   Edit   Selection   View   Go   Run   Terminal   Help

Project2_scripts.sql 2   Results: Project2_scripts.sql ×

Page 1 of 17 |< ◄ ► >| (1-10 of 161 rows)

```
SQL> -- 16) Write a sql query to find out the customers that have at least one sale from Northwest region of America.
SQL> select distinct s.CUSTOMERKEY from sales_2017 s
  2  where s.TERRITORYKEY in (
  3    select t.SALESTERRITORYKEY
  4    from TERRITORIES2 t
  5    where t.REGION='Northwest' and t.SALESTERRITORYKEY=s.TERRITORYKEY
  6  );
```

Select all rows ▼   📋 Save as: CSV ▼ 💾 ⬇ Maximum of 500 rows displayed (modify via MaxRows setting)

CUSTOMERKEY
☑ 24961
☑ 22733
☑ 12180
☑ 11684
☑ 11309
☑ 25979
☑ 23266
☑ 18542
☑ 19718
☑ 23195

Page 1 of 50 |< ◄ ► >| (1-10 of 500 rows)

```
SQL> -- 17) Write a SQL Query to find out which customer has more than one order quantity.
SQL> select * from CUSTOMERS2 c
  2  where c.CUSTOMERKEY in (
  3    select s.CUSTOMERKEY from sales_2017 s
  4    where ORDERQUANTITY>1
  5  );
```

File   Edit   Selection   View   Go   Run   Terminal   Help

Project2_scripts.sql 2   Results: Project2_scripts.sql ×

	CUSTOMERKEY	PREFIX	FIRSTNAME	LASTNAME	BIRTHDATE	MARITALSTATUS	GENDER	EMAILADDRESS	ANNUALINCOME	TOTALCHILDREN	EDUCATIONLEVEL	OCCUPATION	HOMEOWNER	FULL_NAME	SALARY
☑	11067	MR.	CALEB	CARTER	25-SEP-76	S	M	caleb40@adventure-works.com	$60,000	0	Partial College	Professional	Y	MR. CALEB CARTER	60000
☑	11068	MRS.	TIFFANY	LIANG	23-SEP-55	S	F	tiffany17@adventure-works.com	$80,000	2	High School	Skilled Manual	Y	MRS. TIFFANY LIANG	80000
☑	11071	MRS.	LINDA	SERRANO	26-JUN-55	S	F	linda31@adventure-works.com	$80,000	2	High School	Skilled Manual	Y	MRS. LINDA SERRANO	80000
☑	11078	MS.	GINA	MARTIN	10-JAN-74	S	F	gina1@adventure-works.com	$40,000	0	High School	Professional	N	MS. GINA MARTIN	40000
☑	11079	MR.	DONALD	GONZALEZ	11-MAR-59	S	M	donald20@adventure-works.com	$160,000	0	Graduate Degree	Management	N	MR. DONALD GONZALEZ	160000
☑	11082		ANGELA	BUTLER	04-AUG-66	S		angela17@adventure-works.com	$130,000	0	Graduate Degree	Management	N	ANGELA BUTLER	130000
☑	11084	MR.	LUCAS	PHILLIPS	12-SEP-57	S	M	lucas7@adventure-works.com	$80,000	2	Partial High School	Skilled Manual	N	MR. LUCAS PHILLIPS	80000
☑	11085	MRS.	EMILY	JOHNSON	19-JUL-57	S	F	emily1@adventure-works.com	$60,000	2	High School	Professional	N	MRS. EMILY JOHNSON	60000
☑	11087	MRS.	TAMARA	LIANG	03-OCT-57	M	F	tamara6@adventure-works.com	$70,000	3	Partial College	Professional	N	MRS. TAMARA LIANG	70000
☑	11089	MRS.	ABIGAIL	PRICE	05-FEB-57	S	F	abigail25@adventure-works.com	$80,000	2	Bachelors	Management	Y	MRS. ABIGAIL PRICE	80000

Page 1 of 50 |< ◄ ► >| (1-10 of 500 rows)

```
SQL> -- 18) Write a query to find out in which region the following sub category Road Bikes, Mountain Frames are sold and for  which customer. Use CTE
SQL> with pr_table as (
  2  select * from sales_2017 s
  3    join products2 p
  4      on s.PRODUCTKEY=p.PRODUCTKEY),
  5  sc_table as (
  6  select * from products2 p
  7    join PRODUCT_SUBCATEGORIES2 sc
  8      on sc.SUBCATEGORYNAME in ('Road Bikes', 'Mountain Frames') and
  9         p.PRODUCTSUBCATEGORYKEY=sc.PRODUCTSUBCATEGORYKEY)
 10  select * from pr_table
 11  where exists(
 12    select 'pid' from sc_table
 13    where pr_table.productkey=sc_table.productkey
 14  );


    where pr_table.productkey=sc_table.productkey
          *
ERROR at line 91:
ORA-00904: "PR_TABLE"."PRODUCTKEY": invalid identifier
SQL> -- 19) Write a SQL Query to find out which products were returned.
SQL> select p.productkey,p.productname from products2  p
  2  where  not exists (
  3    select 'pkey' from returns2 r
  4    where p.productkey=r.PRODUCTKEY
  5  )order by 1;
```

⊗ 2 ⚠ 0   🌐 Go Live

⊞ 🔍 🔵 🟧 🟢 ▶ 📋 📊 🔷 🟦 🟥 🔵 🟧   ∧ 📶 ⬚ ENG 21:10 20-11-2022

Select all rows ▼   📋 Save as: CSV ▼ 💾 ⬇

PRODUCTKEY	PRODUCTNAME
☑ 218	Mountain Bike Socks, M
☑ 219	Mountain Bike Socks, L
☑ 238	HL Road Frame - Red, 62
☑ 241	HL Road Frame - Red, 44
☑ 244	HL Road Frame - Red, 48
☑ 247	HL Road Frame - Red, 52
☑ 250	HL Road Frame - Red, 56
☑ 253	LL Road Frame - Black, 58
☑ 256	LL Road Frame - Black, 60
☑ 259	LL Road Frame - Black, 62

Page 1 of 17 |< ◄ ► >| (1-10 of 169 rows)

```
SQL> -- 21) Write a SQL Query to find get a report for the following
SQL> -- a) List of all customers
SQL> -- b) Sales done by each customer
SQL> -- c) Product owned by each customer
SQL> -- d) Name of the Product Sub category
SQL> -- e) Products, which were returned.
SQL> with all_prod as(
  2  select *  from customers2
  3    join SALES_2017 using(customerkey)
  4    join products2 p using(productkey)
  5    join PRODUCT_SUBCATEGORIES2 sc
  6      on p.PRODUCTSUBCATEGORYKEY=sc.PRODUCTSUBCATEGORYKEY)
  7  select FULL_NAME,PRODUCTNAME,SUBCATEGORYNAME,
  8    case when productkey in (
  9      select productkey from returns2
 10    ) then 'Returned' else NULL end as "Return Status"
 11  from all_prod;
```

⊗ 2 ⚠ 0   🌐 Go Live

⊞ 🔍 🔵 🟧 🟢 ▶ 📋 📊 🔷 🟦 🟥 🔵 🟧   ∧ 📶 ⬚ ENG 21:10 20-11-2022

Select all rows ▼ | Save as: CSV

FULL_NAME	PRODUCTNAME	SUBCATEGORYNAME	Return Status
MR. FRANK VAZQUEZ	Road Bottle Cage	Bottles and Cages	Returned
MR. FRANK VAZQUEZ	Hydration Pack - 70 oz.	Hydration Packs	Returned
MR. FRANK VAZQUEZ	Touring-1000 Yellow, 60	Touring Bikes	Returned
MR. GARRETT SANDERS	Mountain Bottle Cage	Bottles and Cages	Returned
MR. GARRETT SANDERS	Fender Set - Mountain	Fenders	Returned
MR. GARRETT SANDERS	Water Bottle - 30 oz.	Bottles and Cages	Returned
MR. GARRETT SANDERS	Patch Kit/8 Patches	Tires and Tubes	Returned
MS. GRACE COOK	Sport-100 Helmet, Blue	Helmets	Returned
MS. GRACE COOK	Road-350-W Yellow, 40	Road Bikes	Returned
MRS. GRACE PRICE	Mountain Tire Tube	Tires and Tubes	Returned

Page 1 of 50 |< ◄ ► >| (1-10 of 500 rows)

```
SQL> -- 22) Write a SQL Query using Sub-select to get the count of all table.
SQL> select count(*) as CustomerCount from(
2    select * from customers2);
```

Select all rows ▼ | Save as: CSV

CUSTOMERCOUNT
18148

Page 1 of 1 |< ◄ ► >| (1-1 of 1 rows)

```
SQL> select count(*) as SalesCount from(
2    select * from SALES_2017);
```

SALESCOUNT
29481

Page 1 of 1 |< ◄ ► >| (1-1 of 1 rows)

```
SQL> select count(*) as ProductCount from(
2    select * from products2);
```

Select all rows ▼ | Save as: CSV

PRODUCTCOUNT
293

Page 1 of 1 |< ◄ ► >| (1-1 of 1 rows)

```
SQL> select count(*) as CategoryCount from(
2    select * from PRODUCT_CATEGORIES2);
```

Select all rows ▼ | Save as: CSV

CATEGORYCOUNT
4

Page 1 of 1 |< ◄ ► >| (1-1 of 1 rows)

```
SQL> select count(*) as SCategoryCount from(
2    select * from PRODUCT_SUBCATEGORIES2);
```

Select all rows ▼ | Save as: CSV

SCATEGORYCOUNT
37

Page 1 of 1 |< ◄ ► >| (1-1 of 1 rows)

Go Live   ⊗ 2 ⚠ 0   21:10 ENG 20-11-2022

File Edit Selection View Go Run Terminal Help   Results: Project2_scripts.sql - scripts - Visual Studio Code [Administrator]

Project2_scripts.sql 2   Results: Project2_scripts.sql ✕

SCATEGORYCOUNT
37

Page 1 of 1 |< ◄ ► >| (1-1 of 1 rows)

```
SQL> select count(*) as TerritoryCount from(
2    select * from TERRITORIES2);
```

Select all rows ▼ | Save as: CSV

TERRITORYCOUNT
10

Page 1 of 1 |< ◄ ► >| (1-1 of 1 rows)

```
SQL> select count(*) as ReturnsCount from(
2    select * from returns2);
```

Select all rows ▼ | Save as: CSV

RETURNSCOUNT
1809

Page 1 of 1 |< ◄ ► >| (1-1 of 1 rows)

```
SQL> -- 23) Write a SQL Query to find out which customer has 3rd highest salary using common table expression.
SQL> with sal_table as (
2    select distinct salary,dense_rank() over(order by salary desc) as myrank
3    from customers2
4    ) select c.FULL_NAME,c.salary from sal_table s
5    inner join customers2 c
6        on s.salary=c.salary and  myrank=3;
```

Select all rows ▼ | Save as: CSV

FULL_NAME	SALARY
MRS. ROBIN ALVAREZ	150000
MRS. DANIELLE REED	150000
MR. SETH PHILLIPS	150000
MR. KYLE SCOTT	150000

Go Live   ⊗ 2 ⚠ 0   21:10 ENG 20-11-2022

```sql
-- 2) Create a new column in Customer Table and as Full Name and let it have values from
Prefix, First Name & Last Name.
alter table customers2 add Full_name varchar2(50);
update customers2
set Full_name = PREFIX||' '||FIRSTNAME||' '||LASTNAME;


-- 4) Replace the ($ , ) values from Annual Income and put the values in a new column
that is Salary as numeric field.
alter table customers2 modify salary number(10);
update customers2
set salary = replace(substr(annualincome,2),',','');



-- 6) Give Bonus to the following customer occupation. For other O
-- Professional 50000
-- Clerical 10000
-- Management   25000
-- Manual   2000
update customers2
set salary= case when occupation ='Professional' then salary+50000
    when occupation ='Clerical' then salary+10000
    when occupation ='Management' then salary+25000
    when occupation ='Manual' then salary+2000;

-- 15) Adventure works have decided to change the product colour for a few of their
products along which with their product size. Following is the information.
    -- COLOR,Newcolor
    -- RED,BLACK
    -- NA,BLUE
    -- MULTI,YELLOW
        -- PRODUCT SIZE,NEW SIZE
        -- 0,LARGE
        -- XL,MEDIUM
        -- ALL OTHERS,SMALL
alter table products2 add Newcolor varchar2(10);
alter table products2 add Newsize varchar2(5);
update PRODUCTS2
set Newcolor= case when productcolor='Red' then 'Black'
                when productcolor='NA' then 'Blue'
                when productcolor like '%/%' then 'Yellow'
                else productcolor end;
update PRODUCTS2
set Newsize= case when productsize='0' then 'L'
                when productsize='XL' then 'M'
                else 'S' end;


-- 20) Write a query to add a new column in customers table as username and get the
values from email field. Fetch all the values before @ symbol. Update the new field with
the values populated your query.
alter table customers2 add username varchar(20);
update customers2
set username= substr(EmailAddress,1,(instr(EmailAddress,'@',1)-1));


-- 24) Write a query to replace the Gender value NA to Null.
update customers2
set gender= case when gender='NA' then NULL else gender end;

-- 25) Give the following syntax(Need all the syntax we can do with Alter statement)
-- ● Alter
-- ● Delete
-- ● Update
-- ● Create
-- ● Insert

-- 26) Full form of SQL
```

```
65    The full form of SQL is Structured Query Language
66
67    -- 27) How to apply Primary Key & Foreign Key using Alter statement
68    You can apply primary and foreign key using cmds
69    1.alter table table_name add primary key(col_name)
70    2.alter table table_name add constraint constraint_name foreign key (col_name) references
      (primary_key col_name from parent table)
71
72    -- 28) Share all your scripts which you used to define relationship to create the above
      mentioned database.
73    alter table Customers2 add primary key (CUSTOMERKEY);
74    alter table Products2 add primary key(PRODUCTKEY);
75    alter table Product_Subcategories2 add primary key(PRODUCTSUBCATEGORYKEY);
76    alter table PRODUCT_CATEGORIES2 add primary key(PRODUCTCATEGORYID );
77    alter table Territories2 add primary key(SALESTERRITORYKEY);
78    alter table products2 modify productkey number(5);
79
80    alter table products2 add foreign key(productsubcategory) references
      product_subcategoies2(productsubcategory);
81    alter table returns2 add foreign key(territorykey) references territories2(
      salesterritorykey);
82    alter table returns2 add foreign key(productkey) references products2(productkey);
83    alter table sales_2017 add foreign key(productkey) references products2(productkey);
84    alter table sales_2017 add foreign key(customerkey) references customers2(customerkey);
85    alter table sales_2017 add foreign key(territorykey) references territories2(
      salesterritorykey);
86    alter table product_subcategoies2 add foreign key(productsubcategorykey) references
      product_categoies2(productsubcategoryid);
```