Adventure Cycling Association

**Oracle Database Driven Project**

**Name: Ratnesh Kumar Sah**

**Date: 22-10-2022**

Changes made in Table Names from original file,

|  |  |  |
| --- | --- | --- |
| **Sr no.** | **Original Table Name** | **Revised Table Name** |
| 1. | Calendars | Calendars |
| 2. | Customers | Customers |
| 3. | Products | Products |
| 4. | Category | ProductCategory |
| 5. | Product Subcategory | ProductSubcategory |
| 6. | Sales | Sales |
| 7. | Territories | Territories |
| 8. | Returns | Returns |

**Changes made in Column Names and datatypes,**

1.

|  |  |  |
| --- | --- | --- |
| **Table Name** | **Initial Column Name** | **Revised Column Name** |
| Calendars | Date | C\_Date |

2.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Table Name** | **Column Name** | **Datatype** | **Size (datatype)** | **Revised Size**  **(datatype)** |
| Products | ModelName | Varchar2 | 26 | 27 |

**Adventure Cycling Association** is a non-profit member organization focused on [travel by bicycle (bicycle touring)](https://en.wikipedia.org/wiki/Bicycle_touring). Headquartered in [Missoula](https://en.wikipedia.org/wiki/Missoula), Montana, Adventure Cycling develops [cycling routes](https://en.wikipedia.org/w/index.php?title=Cycling_route&action=edit&redlink=1), publishes maps, provides guided trips, and advocates for better and safer [cycling](https://en.wikipedia.org/wiki/Cycling) in the U.S. The organization grew from a mass cross-country bicycle ride in 1976 to celebrate the [U.S. Bicentennial](https://en.wikipedia.org/wiki/United_States_Bicentennial). Adventure Cycling also publishes a magazine, [Adventure Cyclist](https://en.wikipedia.org/wiki/Adventure_Cyclist_magazine).

Adventure Cycling celebrated its 40th anniversary in 2016 by hosting the Montana Bicycle Celebration in Missoula, promoting events like Bike Your Park Day and Bike Travel Weekend, and publishing its first-ever coffee table book, America's Bicycle Route: The Story of the Trans America Bicycle Trail.

**Origins:** Adventure Cycling Association was founded in 1973 as [Bike centennial](https://en.wikipedia.org/wiki/Bikecentennial) by Dan and Lys Burden and Greg and June Spiel during the couples’ Hemi Stour bicycle ride from Anchorage, Alaska, to Tierra del Fuego, Argentina. They planned Bike centennial as a cross-country bicycle ride to celebrate the bicentennial of the United States. More than 4,100 cyclists took part in the event, riding all or part of the coast-to-coast, 4,250 miles (6,840 km) [Trans America Bicycle Trail](https://en.wikipedia.org/wiki/TransAmerica_Bicycle_Trail) during the summer of 1976. Once the event was completed, Bike centennial lived on as a [501](https://en.wikipedia.org/wiki/501(c)_organization) non-profit member organization to serve the needs of traveling cyclists, developing more bicycle routes and publishing maps. In 1974, Bike centennial began publishing Bike Report magazine, which later became Adventure Cyclist when the organization changed its name to Adventure Cycling Association in 1993.

**Overview:** Adventure Cycling's mission is "to inspire, empower, and connect people to travel by bicycle." Its membership has grown to over 50,000 members, and its route network is one of the largest in the world, encompassing over 50,000 miles (80,000 km).[[3]](https://en.wikipedia.org/wiki/Adventure_Cycling_Association#cite_note-:0-3) Adventure Cycling is also leading the development of the [U.S. Bicycle Route System](https://en.wikipedia.org/wiki/United_States_Bicycle_Route_System), which when complete will offer more than 50,000 miles (80,000 km) of routes for cyclists.

Adventure Cycling's headquarters are located in Missoula, Montana, in the northern Rocky Mountains. Many traveling cyclists make a point of riding through Missoula, where they stop at Adventure Cycling to enjoy free ice cream, catch up on email, ask advice, and get a tour of the building. Many cyclists also have their photos taken, and some end up in Adventure Cycling's National Bicycle Touring Portrait Collection.

The overall goal is to increase the sales and keep a track of the inventory of Sales, Products, and Products Categories & Subcategories. We have seen a spike in our customers Enrolment. As a part of the database management, we would like to perform certain operations on the historic data to understand the sales for the year of 2015.

**Tables:**

1. Calendars - **DONE**
2. Customers - **DONE**
3. Products - **DONE**
4. Product Subcategory- **DONE**
5. ProductCategory- **DONE**
6. Sales - **DONE**
7. Territories- **DONE**
8. Returns- **DONE**

**Note:** You will be creating data based on the data that is provided to you. Along which you will need to apply the relations based on the following information that will shared to you.

**Fields Information:**

**Customers**: This table contains information about all the customers who have purchased goods from Adventures.

* **CustomerKey: This is the Unique key in Customer Table - PK**
* **Prefix: A title used before a person's name.**
* **First Name: Name of the customer.**
* **Last Name: Last name of the customer.**
* **Birth Date: Customer DOB information.**
* **Marital Status: If customer is married or single.**
* **Gender: Gender of customer**
* **Email Address: Customer personal email address**
* **Annual Income: Annual Income of the customer.**
* **Total Children: No of children in the family.**
* **Education Level: Education information of the customer.**
* **Occupation: Type of job done by the customer.**
* **Home Owner: If the customer own any property of its own.**

PRODUCT CATEGORY:

* Product Category Id: It is the Unique ID of the table. **- PK**
* Product Name: Name of the product.

PRODUCTS:

* ProductKey: It’s a unique key of a product. **- PK**
* ProductSubcategoryKey: It is referencing the column Product Subcategory Key from Product Sub Category Table. **- FK**
* ProductSKU: Product sudo name
* ProductName : Product Name Information
* ModelName : Model Name of the product
* ProductDescription: Description of the product developed.
* ProductColor: Color of the product
* ProductSize: Size of the product
* ProductStyle: Product Style information
* ProductCost: Cost of the product
* ProductPrice : Price for the product.

PRODUCT SUB CATEGORY:

* PRODUCTSUBCATEGORYKEY: It is the unique of the table. **- PK**
* SubcategoryName : Defines the Sub category name of the product
* Product Category Key: It refers to the column Product Category Id from product category table(Foreign Key) **- FK**

RETURNS:

* RETURNS: The day when the product was return.
* TERRITORYKEY: Refers to the column SalesTerritoryKey in Work Territories Tables.**FK**
* ProductKey: Refer to the key from products table. **- FK**
* RETURN QUANTITY: Refers to the Quantity of product returned

SALES:

* **OrderDate: The day order was placed.**
* **StockDate: Refers to the stock present in warehouse for the given date.**
* **OrderNumber: Order number for each order to be shipped to customers.**
* **ProductKey : Refers to the column Product Key from the product table. - FK**
* **CustomerKey: Refers to the column Customer Key from the Customer table. - FK**
* **TerritoryKey: Refers to the column SALES Territory Key from the Work Territories Table. FK**
* **OrderLineItem: On what production line the product was built.**
* **OrderQuantity: Quantity of the product ordered.**

**TERRITORIES:**

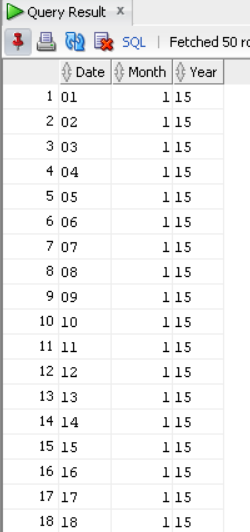
* SALESTERRITORYKEY: It is the Unique key defined for each region. **- PK**
* Region: Directions information
* Country: Refers to which country the product or sales are accomplished.
* Continent: Basic Information.

1. Extract the Month, Day & Year In three different columns in Calendar Table. If Table not created please create the table based on the file received.

**ANS:**  
**select SUBSTR(c\_date,1,2) "Date",**

**extract (month from c\_date) "Month",**

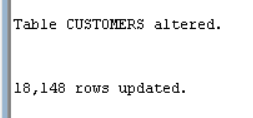
**SUBSTR(c\_date,7,2) "Year" from calendars;**

****

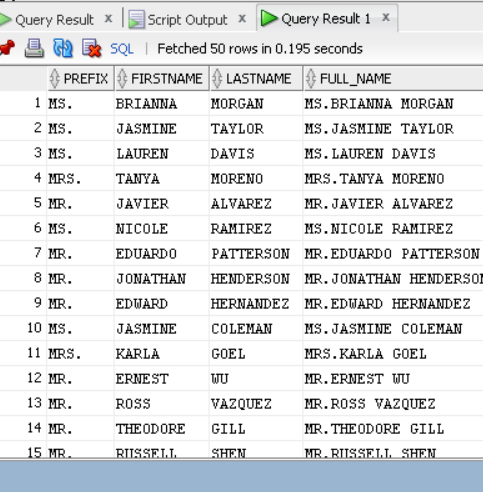
1. Create a new column in Cutomer Table as Full Name and let it have values from Prefix, First Name & Last Name.

**ANS:  
alter table customers ADD FULL\_NAME varchar2(40);**

**update customers set full\_name = prefix || firstname || ' ' || lastname;**



**Select Prefix, FirstName, LastName, Full\_name from Customers;**

****

1. Write a query to find out the number of customer who are married.

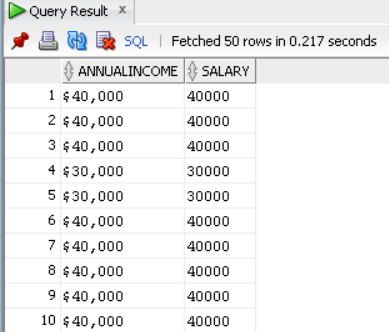
**ANS:**

**select count(\*) from customers where maritalstatus='M';**

****

1. Replace the ($ , ) values from Annual Income and put the values in a new column that is Salary as numeric field.

**ANS:  
select annualincome,replace(replace(annualincome,'$'),',') as salary from customers;**

****

1. Write a query to find out how many customers have 0 kids.

**ANS:  
select count(\*) from customers where totalchildren=0;**

****

1. Give Bonus to the following customer occupation. For other O

|  |  |
| --- | --- |
| Professional | 50000 |
| Clerical | 10000 |
| Management | 25000 |
| Manual | 2000 |

**ANS:**  
 **select case when occupation='Professional' then 50000**

**when occupation='Clerical' then 10000**

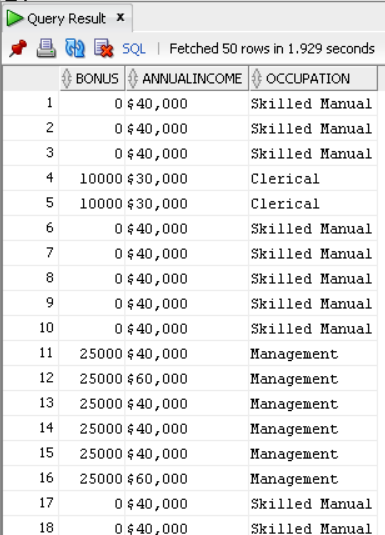
**when occupation='Management' then 25000**

**when occupation='Manual' then 2000**

**else 0**

**end as bonus, annualincome,occupation**

**from customers c;**

****

1. Give me a count of customers who have their own property.

**ANS:  
select count(\*) from customers where homeowner = 'Y';**

****

1. 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.

**ANS:  
select firstname,lastname from customers**

**where firstname like '%DA' AND LASTNAME like 'RA%'**

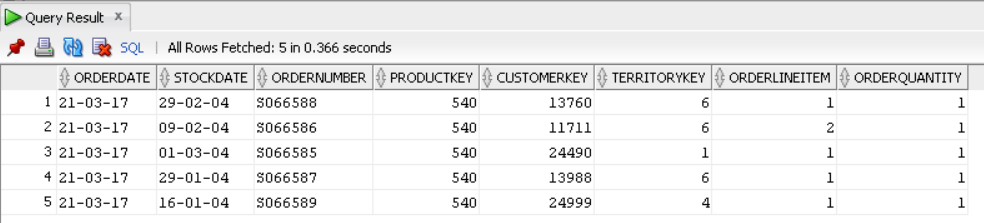
**group by (lastname,firstname) having count(\*) <=1;**



1. Write a query to display the sales for the order date 03/21/2017 for product key 540.

**ANS:  
select \* from sales**

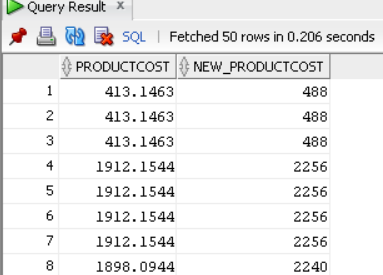
**where orderdate='21-MAR-2017' and productkey='540';**



1. Write a SQL Query to increase the cost of products by 18% and round the data to the nearest number.

**ANS:  
select productcost,round(productcost+(0.18\*productcost)) as new\_productcost**

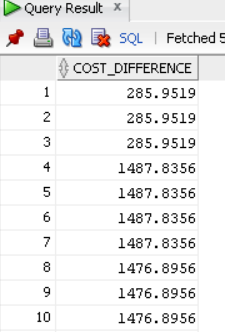
**from products;**



1. Adventure work Head of sales would like to find out the cost difference between productcost and product price.

**ANS:  
select productprice-productcost as cost\_difference**

**from products;**

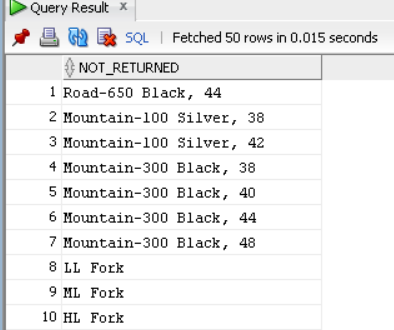


1. Write a SQL Query to find out, which products were not, returned (Use tables Product & Returns) solve the query without ‘not in’ function.

**ANS:  
select productname as NOT\_RETURNED from products where productkey in**

**(select productkey from products minus**

**select productkey from returns);**



1. Write a query to find out which customer has placed most number of sales.

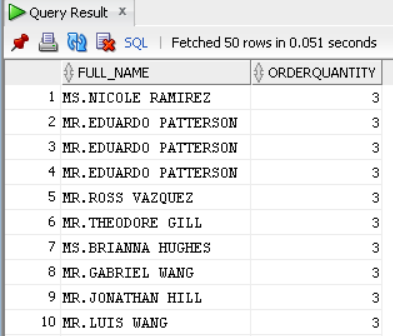
**ANS:  
  
SELECT Full\_name,S.orderquantity**

**FROM CUSTOMERS C**

**LEFT OUTER JOIN SALES S**

**ON C.CUSTOMERKEY=S.CUSTOMERKEY**

**where s.orderquantity in (select max(orderquantity) from sales);**

****

1. Write a SQL Query to find out the products returned for Region Germany.

**ANS:  
select R.productkey,**

**P.productname,**

**T.Region**

**from returns R**

**left outer join products P**

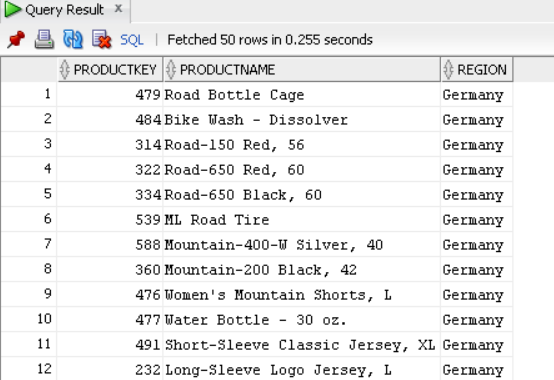
**on R.productkey=P.productkey**

**left outer join territories T**

**on R.territorykey=T.salesterritorykey**

**where T.Region='Germany'**

**group by R.productkey,P.productname,T.Region;**

****

1. 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 | New color |
| RED | BLACK |
| NA | BLUE |
| MULTI | YELLOW |

|  |  |
| --- | --- |
| PRODUCT SIZE | NEW SIZE |
| 0 | LARGE |
| XL | MEDIUM |
| ALL OTHERS | SMALL |

**ANS:  
select case**

**when productcolor='Red' then 'Black'**

**when productcolor='NA' then 'Blue'**

**when productcolor='Multi' then 'Yellow'**

**else productcolor**

**end as n\_color,productcolor,**

**case**

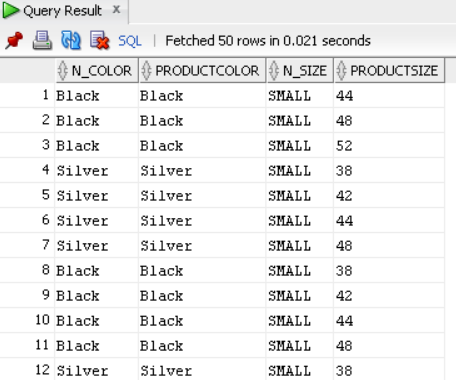
**when productsize='0' then 'LARGE'**

**when productsize='XL' then 'MEDIUM'**

**else 'SMALL'**

**end as n\_size,productsize**

**from products;**

****

1. Write a sql query to find out the customers that have at least one sale from Northwest region of America.

**ANS:  
select C.customerkey,**

**C.firstname,**

**C.lastname,**

**S.territorykey, region,**

**COUNT(C.CUSTOMERKEY) AS SALES**

**from customers C**

**left outer join sales S**

**on C.customerkey=S.customerkey**

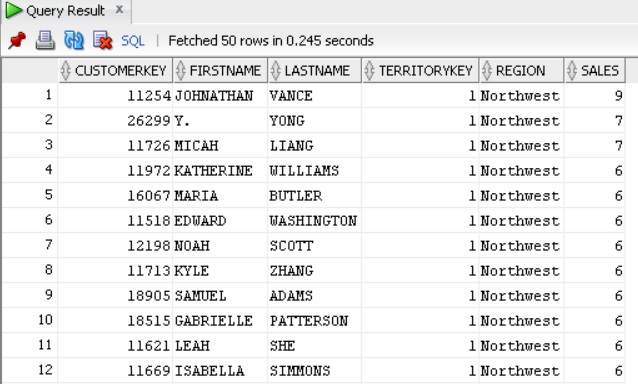
**left outer join territories T**

**on S.territorykey=T.salesterritorykey**

**where region='Northwest'**

**group by C.customerkey,C.firstname,C.lastname,s.territorykey, region**

**order by sales desc;**

****

1. Write a SQL Query to find out which customer has more than one order quantity.

**ANS:  
select c.FIRSTNAME,c.LASTNAME,c.customerkey,s.orderquantity**

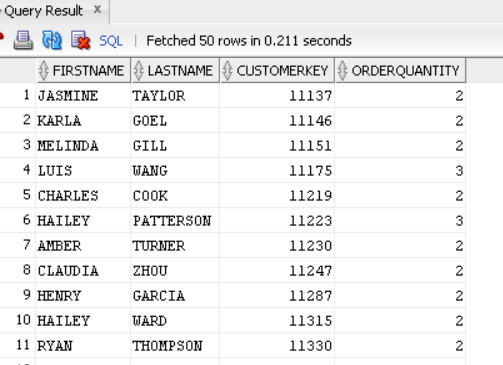
**from customers c**

**left outer join sales s**

**on c.customerkey=s.customerkey**

**where orderquantity > 1**

**group by c.firstname,c.lastname,c.customerkey, s.orderquantity;**

****

1. Write a query to find out in which region the following sub category Road Bikes, Mountain Frames are sold and by which customer. Use CTE

**ANS:  
 with res as**

**(select c.customerkey as custkey,**

**c.firstname as fname,**

**c.lastname as lname**

**from customers c),**

**vi as**

**(select s.customerkey as customerkey,**

**s.productkey as productkey,**

**t.region as region,**

**t.salesterritorykey,**

**s.territorykey**

**from sales s**

**inner join territories t**

**on s.territorykey = t.salesterritorykey),**

**x as**

**(select ps.productkey as prodkey,**

**ps.productsubcategorykey as productsubcategorykey**

**from products ps) ,**

**v as**

**(select p.productsubcategorykey as pskey,**

**p.subcategoryname as name**

**from productsubcategory p)**

**select res.fname,res.lname,vi.region, res.custkey, vi.customerkey, vi.productkey ,x.prodkey, x.productsubcategorykey, v.name**

**from res**

**inner join vi**

**on res.custkey = vi.customerkey**

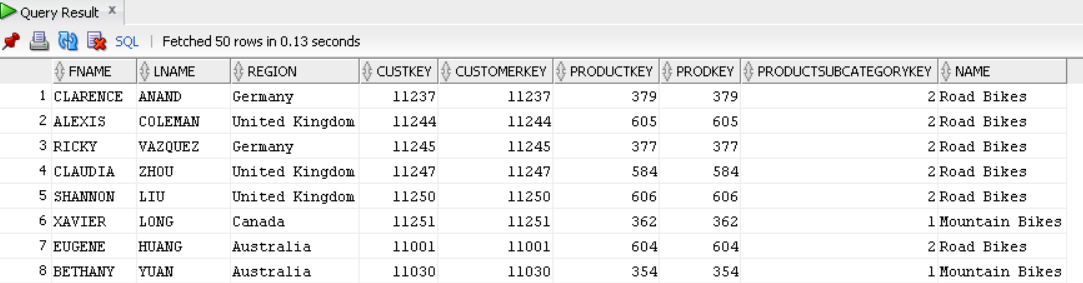
**inner join x**

**on vi.productkey = x.prodkey**

**inner join v**

**on x.productsubcategorykey = v.pskey**

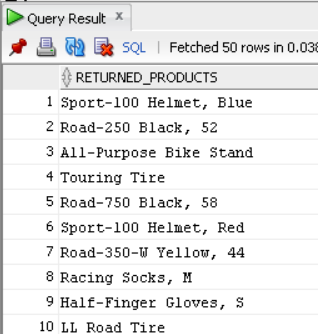
**where v.name like '%Mountain Bikes%' or v.name like '%Road Bike%';**



1. Write a SQL Query to find out which products were returned.

**ANS:  
select productname as RETURNED\_PRODUCTS from products**

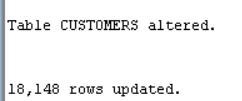
**where productkey in (select productkey from returns);**

****

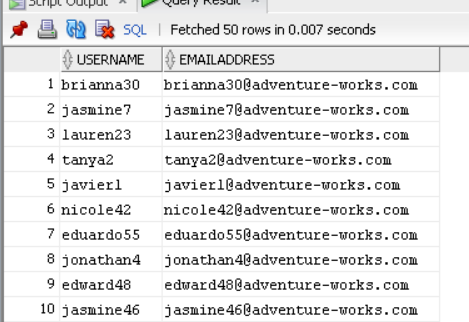
1. Write a query to add a new column in customers table as username and get the values from email field. Fetch all the values before @ symbol. Update the new field with the values populated your query.

**ANS:**  
**ALTER TABLE CUSTOMERS ADD USERNAME VARCHAR2(40);**

**UPDATE CUSTOMERS SET USERNAME=SUBSTR(EMAILADDRESS,0,INSTR(EMAILADDRESS,'@')-1);**

****

**SELECT USERNAME, EMAILADDRESS FROM CUSTOMERS ;**

****

1. Write a SQL Query to find get a report for the following
2. List of all customers
3. Sales done by each customer
4. Product owned by each customer
5. Name of the Product Sub category
6. Products, which were returned.

**ANS:  
WITH A\_CTE AS**

**(SELECT T1.CUSTOMERKEY AS cust\_customerkey,**

**T1.FULL\_NAME AS CUST\_FULL\_NAME**

**FROM CUSTOMERS T1),**

**B AS**

**(SELECT T2.PRODUCTKEY AS PRODUCTKEY\_SALES,**

**T2.CUSTOMERKEY AS SALES\_CUSTOMERKEY**

**FROM SALES T2),**

**C AS**

**(SELECT T3.PRODUCTKEY AS PRODUCTKEY\_PRODS,**

**T3.PRODUCTSUBCATEGORYKEY AS PROD\_SUB\_CAT\_KEY\_PRODS,**

**T3.PRODUCTNAME AS PRODUCT\_NAME\_PRODS,**

**T4.PRODUCTKEY AS PRODUCTKEY\_RETURN**

**FROM PRODUCTS T3**

**LEFT OUTER JOIN RETURNS T4**

**ON T3.PRODUCTKEY=T4.PRODUCTKEY),**

**D AS**

**(SELECT T5.PRODUCTSUBCATEGORYKEY AS PROD\_SUB\_CAT\_KEY\_PRODSUBCAT,**

**T5.SUBCATEGORYNAME AS PROD\_SUB\_CAT\_NAME**

**FROM productsubcategory T5)**

**SELECT DISTINCT CUST\_FULL\_NAME,PRODUCT\_NAME\_PRODS,PRODUCTKEY\_RETURN,PROD\_SUB\_CAT\_NAME FROM A\_CTE**

**LEFT OUTER JOIN B**

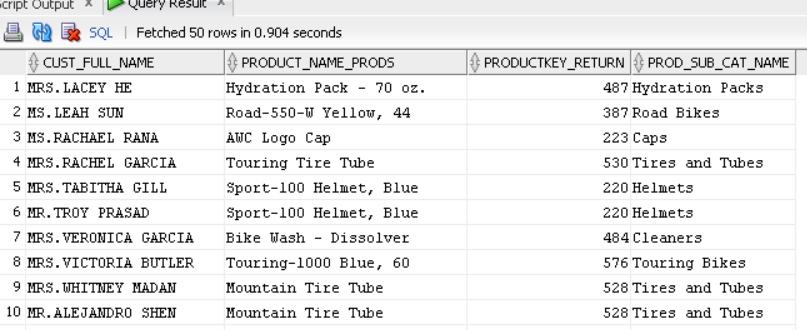
**ON A\_CTE.cust\_customerkey = B.SALES\_CUSTOMERKEY**

**LEFT OUTER JOIN C**

**ON B.PRODUCTKEY\_SALES = C.PRODUCTKEY\_PRODS**

**LEFT OUTER JOIN D**

**ON C.PROD\_SUB\_CAT\_KEY\_PRODS = D.PROD\_SUB\_CAT\_KEY\_PRODSUBCAT;**



1. Write a SQL Query using Sub-select to get the count of all table.

**ANS:  
select count(\*) from**

**(SELECT T1.CUSTOMERKEY AS cust\_customerkey,**

**T1.FULL\_NAME AS CUST\_FULL\_NAME**

**FROM CUSTOMERS T1) A**

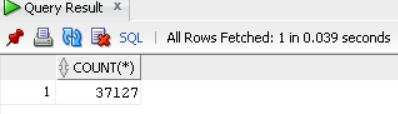
**LEFT OUTER JOIN**

**(SELECT T2.PRODUCTKEY AS PRODUCTKEY\_SALES,**

**T2.CUSTOMERKEY AS SALES\_CUSTOMERKEY**

**FROM SALES T2) B**

**ON A.cust\_customerkey = B.SALES\_CUSTOMERKEY;**



1. Write a SQL Query to find out which customer has 3rd highest salary using common table expression.

**ANS:  
WITH A\_CTE AS**

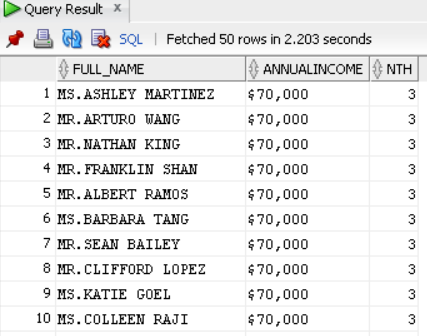
**(SELECT full\_name,annualincome,**

**DENSE\_RANK() OVER (ORDER BY annualincome Desc) AS nth**

**FROM customers)**

**SELECT \* FROM A\_CTE**

**WHERE nth=3;**



1. Write a query to replace the Gender value NA to Null.

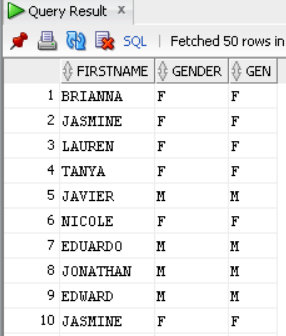
**ANS:  
select firstname,gender,**

**case when gender = 'NA' then null**

**else gender**

**end as gen**

**from customers;**



1. Give the following syntax(Need all the syntax we can do with Alter statement)

* Alter
* Delete
* Update
* Create
* Insert

**ANS:**  
 **Alter: -**

alter table table\_name add constraint constraint\_name key\_name(column\_name);

alter table table\_name add column\_name datatype(size);

alter table table\_name add column\_name datatype(size),column\_name datatype(size);

alter table table\_name drop column column\_name;

alter table table\_name modify column column\_name datatype ;

alter table table\_name rename to new\_table\_name;

alter table table\_name rename column column\_name to new\_column\_name;

alter table table\_name alter column column\_name type varchar(25);

alter table table\_name drop constraint constraint\_name;

**Delete: -**

delete from table;

delete from table where eno=10;

**Update: -**

update table Set colum-1= value-1,....,column-n=value-n

**Create: -**

create table table\_name(column\_name datatype(size));

**Insert: -**

insert into table values('value1','value2');

1. Full form of SQL

**ANS:**  
 **Structured Query Language**

1. How to apply Primary Key & Foreign Key using Alter statement

**ANS:**  
 **ALTER TABLE table\_name ADD PRIMARY KEY (column\_name);**

**ALTER TABLE table\_name ADD CONSTRAINT fk\_foreign key  
 (column\_name) REFRENCES table\_name2(column\_name);**

1. Share all your scripts which you used to define relationship to create the above mentioned database.

**ANS:**  
 **1. CUSTOMERS**

**- SELECT \*FROM CUSTOMERS;**

**- ALTER TABLE CUSTOMERS ADD PRIMARY KEY(CUSTOMERKEY);**

**2. PRODUCTCATEGORY**

**- SELECT \*FROM PRODUCTCATEGORY;**

**- ALTER TABLE PRODUCTCATEGORY ADD PRIMARY KEY(PRODUCTCATEGORYID);**

**3. PRODUCTS**

**- SELECT \*FROM PRODUCTS;**

**- ALTER TABLE PRODUCTS ADD PRIMARY KEY(PRODUCTKEY);**

**- ALTER TABLE PRODUCTS ADD CONSTRAINT FK\_PRODUCTSUBCATEGORYKEY FOREIGN KEY(PRODUCTSUBCATEGORYKEY) REFERENCES PRODUCT\_SUBCATEGORY(PRODUCTSUBCATEGORYKEY);**

**4. PRODUCTSUBCATEGORY**

**- SELECT \*FROM PRODUCTSUBCATEGORY;**

**- ALTER TABLE PRODUCTSUBCATEGORY   
 ADD PRIMARY KEY(PRODUCTSUBCATEGORYKEY);**

**- ALTER TABLE PRODUCTSUBCATEGORY ADD CONSTRAINT FK\_PRODUCTCATEGORYKEY FOREIGN KEY(PRODUCTCATEGORYKEY)   
REFERENCES PRODUCTCATEGORY(PRODUCTCATEGORYID);**

**5. RETURNS**

**- SELECT \* FROM RETURNS;**

**- ALTER TABLE RETURNS ADD CONSTRAINT FK\_PRODUCTKEY   
 FOREIGN KEY(PRODUCTKEY) REFERENCES PRODUCTS(PRODUCTKEY);**

**- ALTER TABLE RETURNS ADD CONSTRAINT FK\_TERRITORYKEY\_RETURNS   
FOREIGN KEY(TERRITORYKEY) REFERENCES TERRITORIES(SALESTERRITORYKEY);**

**- ALTER TABLE RETURNS ADD CONSTRAINT RETURNS\_COMPK PRIMARY KEY (PRODUCTKEY,TERRITORYKEY,RETURNDATE);**

**6. SALES**

**- SELECT \* FROM SALES;**

**- ALTER TABLE SALES ADD CONSTRAINT FK\_PRODUCTKEY\_SALES  
 FOREIGN KEY(PRODUCTKEY) REFERENCES PRODUCTS(PRODUCTKEY);**

**- ALTER TABLE SALES ADD CONSTRAINT FK\_CUSTOMERKEY FOREIGN KEY(CUSTOMERKEY) REFERENCES CUSTOMER(CUSTOMERKEY);**

**- ALTER TABLE SALES ADD CONSTRAINT FK\_TERRITORYKEY   
FOREIGN KEY(TERRITORYKEY) REFERENCES TERRITORIES(SALESTERRITORYKEY);**

**- ALTER TABLE RETURNS ADD CONSTRAINT SALES\_COMPK PRIMARY KEY (PRODUCTKEY,CUSTOMERKEY,ORDERDATE,STOCKDATE);**

**7. TERRITORIES**

**- SELECT \*FROM TERRITORIES;**

**- ALTER TABLE TERRITORIES ADD PRIMARY KEY(SALES TERRITORYKEY);**

**E-R DIAGRAM**

