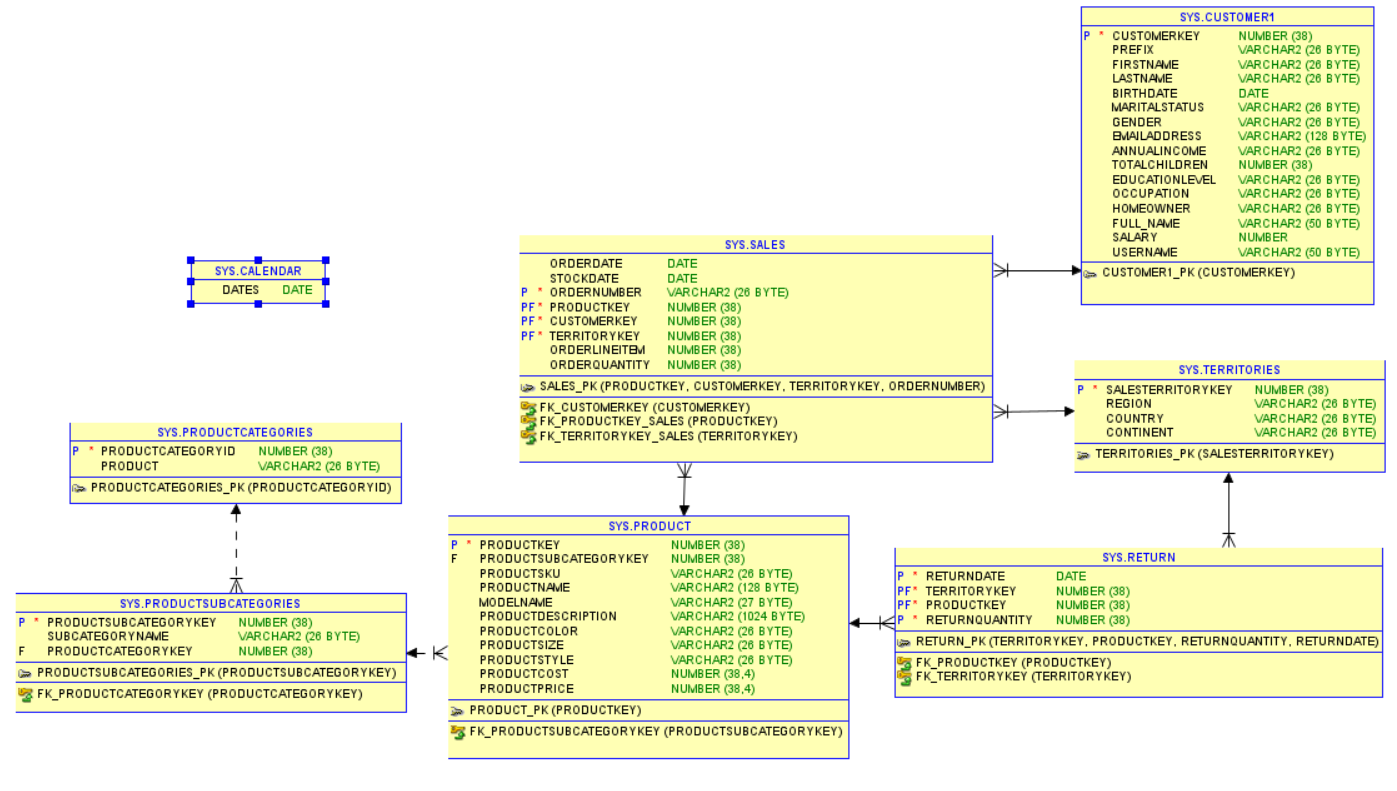
ER Diagram:



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.

Ans: select extract(month from dates),

extract(day from dates),

extract(year from dates)

from calendar;

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

Ans: alter table customer1 add (full\_name varchar2(50));

update customer1

set full\_name = prefix||firstname||' '||lastname;

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

Ans: select count(\*) from customer1 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: alter table customer1 add (salary number);

update customer1

set salary = cast(replace(replace(annualincome,'$',''),',','') as number);

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

Ans: select count(\*) from customer1 where totalchildren = 0;

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

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

Ans: select customerkey, firstname, lastname,

occupation,

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

from customer1;

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

Ans: select count(\*) from customer1 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.

select distinct \* from customer1

where firstname like '%DA'

and lastname like 'RA%';

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

Ans: select \* from sales

where productkey = '540'

and orderdate = '21/3/2017';

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

Ans: select productkey, productcost, round((productcost\*1.18)) as enhanced\_cost

from product;

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

Ans: select productkey, productprice, productcost, productprice-productcost as diff

from product;

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 p.productkey, p.productname

from product p

inner join

(select productkey from (

(select productkey from product) minus

(select productkey from return))) a

on p.productkey = a.productkey;

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

Ans: select c.customerkey, c.firstname, c.lastname

from customer1 c

where customerkey in (

select customerkey from (

select customerkey, rank() over(order by count(\*) desc) as rnk

from sales

group by customerkey

)

where rnk = 1);

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

Ans: select distinct p.productkey, p.productname

from return r

inner join territories t on r.territorykey = t.salesterritorykey

inner join product p on r.productkey = p.productkey

where t.region = 'Germany';

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: update product

set productcolor =

case when Productcolor = 'Red' then 'Black'

when productcolor = 'NA' then 'Blue'

when productcolor = 'Multi' then 'Yellow'

else productcolor

end,

productsize =

case when Productsize = '0' then 'Large'

when productsize = 'XL' then 'Medium'

else 'Small'

end;

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

Ans: select distinct c.customerkey,c.firstname, c.lastname

from sales s

inner join territories t on s.territorykey = t.salesterritorykey

inner join customer1 c on c.customerkey = s.customerkey

where t.region = 'Northwest' and s.orderquantity is not null;

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

Ans: select c.customerkey, c.firstname, c.lastname

from customer1 c

inner join sales s on c.customerkey = s.customerkey

where s.orderquantity > 1;

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 CTE1 as

(

select productsubcategorykey,

subcategoryname

from productsubcategories

),

CTE2 as

(

select productsubcategorykey,

productkey

from product

),

CTE3 as

(

select productkey,

customerkey,

territorykey

from sales

),

CTE4 as

(

select customerkey,

firstname

from customer1

),

CTE5 as

(

select salesterritorykey,

region

from territories

)

select t4.customerkey,

t4.firstname,

t1.subcategoryname,

t5.region

from CTE1 t1

inner join CTE2 t2

on t1.productsubcategorykey=t2.productsubcategorykey

inner join CTE3 t3

on t2.productkey=t3.productkey

inner join CTE4 t4

on t3.customerkey=t4.customerkey

inner join CTE5 t5

on t3.territorykey=t5.salesterritorykey

where t1.subcategoryname in('Road Bikes','Mountain Frames');

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

Ans: select distinct productkey, productname

from product p

where p.productkey in (select distinct productkey from return);

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 customer1 add username varchar2(50);

update customer1

set username = substr(emailaddress,0,instr(emailaddress,'@')-1);

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

Ans: select \* from

(

select 'Customer' as table\_name, count(\*) as count from customer1

union

select 'Sales' as table\_name, count(\*) as count from sales

union

select 'Products' as table\_name, count(\*) as count from product

union

select 'productcategories' as table\_name, count(\*) as count from productcategories

union

select 'Productsubcategories' as table\_name, count(\*) as count from productsubcategories

union

select 'Territories' as table\_name, count(\*) as count from territories

union

select 'Return' as table\_name, count(\*) as count from return

);

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

Ans: with ordered\_salary as(

select customerkey, firstname, lastname, salary, dense\_rank() over(order by salary desc) rn

from customer1

)

select distinct customerkey, firstname, lastname, salary

from ordered\_salary

where rn = 3;

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

Ans: update customer1

set gender = null

where gender = 'NA';

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

* Alter
* Delete
* Update
* Create
* Insert

Ans:

* Alter:

Add column : alter table table\_name add column\_name data-type;

Add Multiple column: ALTER TABLE table\_name ADD (column\_1 column\_definition, column\_n column\_definition);

Modify Column : ALTER TABLE table\_name MODIFY (column\_name1 column\_type…n);

Add Primary key: alter table customer1 add primary key (CUSTOMERKEY);

Drop Column: ALTER TABLE table\_name DROP COLUMN column\_name;

Rename Column: ALTER TABLE table\_name RENAME COLUMN old\_name TO new\_name;

Rename table: ALTER TABLE table\_name RENAME TO new\_table\_name;

Add foreign key: alter table products add constraint fk\_PRODUCTSUBCATEGORYKEY foreign key (PRODUCTSUBCATEGORYKEY) references ProductSubcategories(PRODUCTSUBCATEGORYKEY);

* Delete: delete from table\_name [where condition];
* Update: update table\_name set column\_name = value where condition;
* Create: create table table\_name(col1 data-type,….col-n data-type);
* Insert: insert into table\_name(provide column list) values(value1,value2,….value-n);

1. Full form of SQL

Ans: Structured Query Language

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

Ans:

Syntax for primary key: alter table table\_name add primary key(column-name);

alter table customer1 add primary key (CUSTOMERKEY);

Foreign Key:

Syntax:

alter table table\_name add constraint fk\_give\_constraint\_name foreign key (column name where reference should be applied) references table\_name\_from\_where\_we\_take\_reference(primary\_key\_column\_which\_need\_to\_refer);

alter table products add constraint fk\_PRODUCTSUBCATEGORYKEY foreign key (PRODUCTSUBCATEGORYKEY) references ProductSubcategories(PRODUCTSUBCATEGORYKEY);

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

Ans:

alter table customer1 add primary key (CUSTOMERKEY);

alter table PRODUCTCATEGORIES add primary key (productcategoryid);

alter table PRODUCT add primary key (PRODUCTKEY);

alter table PRODUCTSUBCATEGORIES add primary key (PRODUCTSUBCATEGORYKEY);

alter table TERRITORIES add primary key (SALESTERRITORYKEY);

alter table return add primary key (TERRITORYKEY,PRODUCTKEY,RETURNQUANTITY,RETURNDATE);

alter table sales add primary key (PRODUCTKEY,CUSTOMERKEY,TERRITORYKEY,ORDERNUMBER);

alter table calendar add primary key (DATES);

Foreign Key:

ALTER TABLE PRODUCT

ADD CONSTRAINT FK\_ PRODUCTSUBCATEGORYKEY

FOREIGN KEY (PRODUCTSUBCATEGORYKEY)

REFERENCES PRODUCTSUBCATEGORIES (PRODUCTSUBCATEGORYKEY);

ALTER TABLE PRODUCTSUBCATEGORIES

ADD CONSTRAINT FK\_ PRODUCTCATEGORYSUBKEY

FOREIGN KEY (PRODUCTSUBCATEGORYKEY)

REFERENCES PRODUCTCATEGORIES (PRODUCTCATEGORYID);

ALTER TABLE SALES

ADD CONSTRAINT FK\_ SALES

FOREIGN KEY (PRODUCTKEY)

REFERENCES PRODUCT (PRODUCTKEY);

ALTER TABLE SALES

ADD CONSTRAINT FK\_ SALES\_CUSTOMERKEY

FOREIGN KEY (CUSTOMERKEY)

REFERENCES CUSTOMER1 (CUSTOMERKEY);

ALTER TABLE SALES

ADD CONSTRAINT FK\_ SALES\_TERRITORYKEY

FOREIGN KEY (TERRITORYKEY)

REFERENCES TERRITORIES (SALESTERRITORYKEY);

ALTER TABLE RETURN

ADD CONSTRAINT FK\_ RETURNS

FOREIGN KEY (TERRITORYKEY)

REFERENCES TERRITORIES (SALESTERRITORYKEY);

ALTER TABLE RETURN

ADD CONSTRAINT FK\_ PRODUCTKEY

FOREIGN KEY (PRODUCTKEY)

REFERENCES PRODUCT (PRODUCTKEY);