Date 18-08-2021

* Joins – LEFT JOIN, RIGHT JOIN,,FULL JOIN, FULL OUTER JOIN , INNER JOIN

SELECT \*

FROM hr.candidates c

INNER JOIN hr.employee e

ON e.fullname = c.fullname;

SELECT \*

FROM hr.candidates c

FULL JOIN hr.employee e

ON e.fullname = c.fullname;

SELECT \*

FROM hr.candidates c

FULL OUTER JOIN hr.employee e

ON e.fullname = c.fullname;

SELECT \*

FROM hr.candidates c

RIGHT JOIN hr.employee e

ON e.fullname = c.fullname;

SELECT \*

FROM hr.candidates c

LEFT JOIN hr.employee e

ON e.fullname = c.fullname;

SELECT \*

FROM hr.candidates c

CROSS JOIN hr.employee e

ON e.fullname = c.fullname;

* GROUP BY ,HAVING :-

SELECT ( first\_name )first\_name,

Max(phone) phone\_number

FROM sales.customers c

GROUP BY phone

HAVING phone IS NULL

* CUBE :-

SELECT

brand,

category,

SUM (sales) sales

FROM

sales.sales\_summary

GROUP BY

CUBE(brand, category);

* ROLL UP :-

SELECT

brand,

category,

SUM (sales) sales

FROM

sales.sales\_summary

GROUP BY

ROLLUP(brand, category);

* SUB QUERY :-

SELECT

order\_id,

order\_date,

customer\_id

FROM

sales.orders

WHERE

customer\_id IN (

SELECT

customer\_id

FROM

sales.customers

WHERE

city = 'New York'

)

ORDER BY

order\_date DESC;

* CORRELATED SUBQUERY :

SELECT

product\_name,

list\_price,

category\_id

FROM

production.products p1

WHERE

list\_price IN (

SELECT

MAX (p2.list\_price)

FROM

production.products p2

WHERE

p2.category\_id = p1.category\_id

GROUP BY

p2.category\_id

)

ORDER BY

category\_id,

product\_name;

* Exists: -

SELECT

customer\_id,

first\_name,

last\_name

FROM

sales.customers

WHERE

EXISTS (SELECT NULL)

ORDER BY

first\_name,

last\_name;

* ANY :-

SELECT

product\_name,

list\_price

FROM

production.products

WHERE

product\_id = ANY (

SELECT

product\_id

FROM

sales.order\_items

WHERE

quantity >= 2

)

ORDER BY

product\_name;

* INTERSECT :-

SELECT

city

FROM

sales.customers

INTERSECT

SELECT

city

FROM

sales.stores

ORDER BY

city;

* Creating new database called Online\_shopping\_System
* Creating tables

CREATE TABLE customer

(

customer\_id INT PRIMARY KEY IDENTITY (1, 1),

NAME VARCHAR(255),

contact\_add INT,

address TEXT

);

CREATE TABLE categories

(

category\_id INT PRIMARY KEY IDENTITY (1, 1),

category\_name VARCHAR(255),

category\_type VARCHAR(255)

);

CREATE TABLE shoppingorder

(

order\_id INT PRIMARY KEY IDENTITY (1, 1),

customer\_id INT

CONSTRAINT fk\_group FOREIGN KEY (customer\_id) REFERENCES customer(

customer\_id),

date DATE

);

CREATE TABLE deliveries

(

acc\_id INT PRIMARY KEY IDENTITY (1, 1),

customer\_id INT

CONSTRAINT fk\_deliveries FOREIGN KEY (customer\_id) REFERENCES customer

(

customer\_id),

date DATE

);

CREATE TABLE products

(

product\_id INT PRIMARY KEY IDENTITY (1, 1),

category\_id INT

CONSTRAINT fk\_products FOREIGN KEY (category\_id) REFERENCES categories

(

category\_id),

product\_name VARCHAR(255)

);

CREATE TABLE seller

(

seller\_id INT PRIMARY KEY IDENTITY (1, 1),

product\_id INT

CONSTRAINT fk\_seller FOREIGN KEY (product\_id) REFERENCES products(

product\_id),

seller\_name VARCHAR(255)

);

CREATE TABLE payment

(

payment\_id INT PRIMARY KEY IDENTITY (1, 1),

customer\_id INT

CONSTRAINT fk\_payment FOREIGN KEY (customer\_id) REFERENCES customer(

customer\_id),

date DATE

);

CREATE TABLE transaction\_report

(

report\_id INT PRIMARY KEY IDENTITY (1, 1),

customer\_id INT

CONSTRAINT fk\_transaction FOREIGN KEY (customer\_id) REFERENCES

customer(

customer\_id),

order\_id INT

CONSTRAINT fk\_order FOREIGN KEY (order\_id) REFERENCES shoppingorder(

order\_id),

product\_id INT

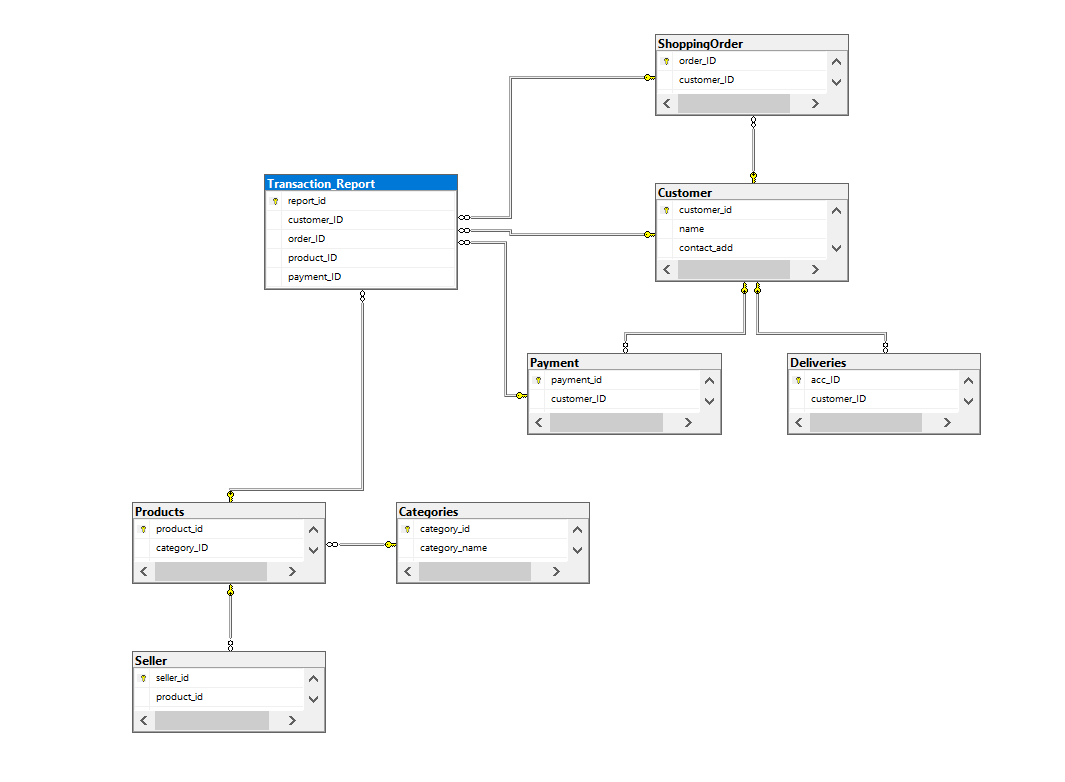
CONSTRAINT fk\_pro FOREIGN KEY (product\_id) REFERENCES products(

product\_id),

payment\_id INT

CONSTRAINT fk\_pay FOREIGN KEY (payment\_id) REFERENCES payment(payment\_id)

);

* ER diagram of Online\_shopping\_system :-