SQL ASSIGNMENT 1

1.Design the complete database + schema + tables for the diagram shown above using appropriate data type for every column along with any constraints (checks + PK) mentioned in the task description and load the below data into the requisite tables.

-- CREATE DATABASE

```
CREATE DATABASE BikeStores
```

-- creating schemas (production & sales)

```
CREATE SCHEMA production
CREATE SCHEMA sales
```

---Creating table BIKESTORES.PRODUCTION.CATEGORIES

```
CREATE OR REPLACE TABLE BIKESTORES.PRODUCTION.CATEGORIES(
        category_id INT  PRIMARY KEY,
   category_name VARCHAR (255)
);
```

---Creating table BIKESTORES. PRODUCTION.BRANDS

```
CREATE OR REPLACE TABLE BIKESTORES.PRODUCTION.BRANDS (
        brand_id INT  PRIMARY KEY,
        brand_name VARCHAR (255) ,
  );
```

------Creating table BIKESTORES.PRODUCTION.PRODUCTS

```
CREATE OR REPLACE TABLE BIKESTORES.PRODUCTION.PRODUCTS (
        product_id INT  PRIMARY KEY,
        product_name VARCHAR (255) ,
        brand_id INT ,
        category_id INT ,
        model_year SMALLINT,
        list_price DECIMAL (10, 2) );
```

------Creating table BIKESTORES.PRODUCTION.STOCKS

```
CREATE OR REPLACE TABLE BIKESTORES.PRODUCTION.STOCKS (
        store_id INT,
        product_id INT,
        quantity INT,
        PRIMARY KEY (store_id, product_id)
        );
```

--CREATING TABLES IN SALES BIKESTORES.SALES.CUSTOMERS

```
CREATE OR REPLACE TABLE BIKESTORES.SALES.CUSTOMERS(
        customer_id NUMBER (40,2) ,
        first_name VARCHAR (255) ,
        last_name VARCHAR (255) ,
        phone VARCHAR (25),
        email VARCHAR (255) ,
        street VARCHAR (255),
        city VARCHAR (50),
        state VARCHAR (25),
        zip_code VARCHAR (5),
    primary key (customer_id)
);
```

-----CREATING TABLES BIKESTORES.SALES.STORES

```
CREATE OR REPLACE TABLE BIKESTORES.SALES.STORES(
        store_id INT PRIMARY KEY,
        store_name VARCHAR (255) ,
        phone VARCHAR (25),
        email VARCHAR (255),
        street VARCHAR (255),
        city VARCHAR (255),
        state VARCHAR (10),
        zip_code VARCHAR (5)
);
```

-----CREATING TABLES BIKESTORES.SALES.STAFFS

```
CREATE OR REPLACE TABLE BIKESTORES.SALES.STAFFS (
        staff_id INT  PRIMARY KEY,
        first_name VARCHAR (50) ,
        last_name VARCHAR (50) ,
        email VARCHAR (255) ,
        phone VARCHAR (25),
        active tinyint ,
        store_id INT ,
        manager_id INT
        );
```

--------CREATING TABLES BIKESTORES.SALES.ORDERS

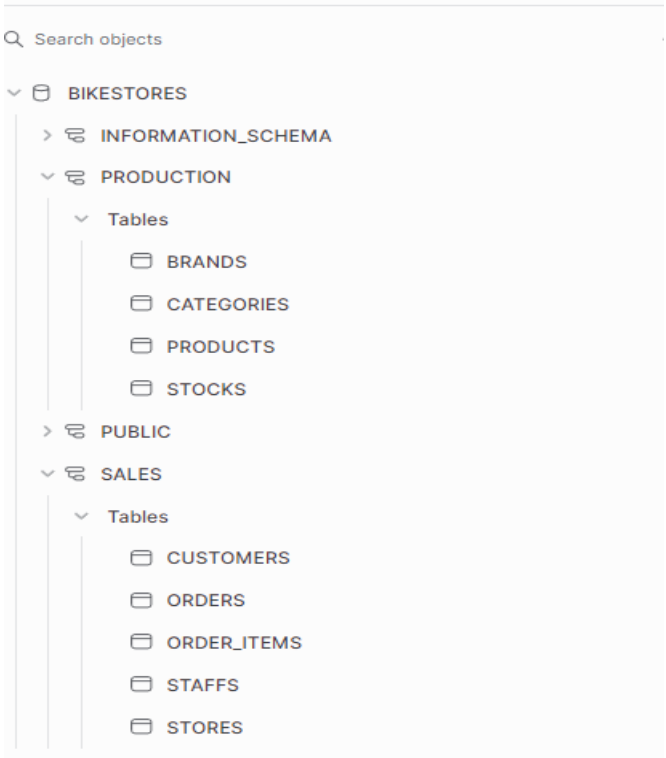
```
CREATE OR REPLACE TABLE BIKESTORES.SALES.ORDERS (
        order_id INT  PRIMARY KEY,
        customer_id INT,
        order_status tinyint ,
        order_date DATE ,
        required_date DATE ,
        shipped_date DATE,
        store_id INT ,
        staff_id INT
        );
```

----------**CREATING TABLES BIKESTORES.SALES.ITEMS**

```
CREATE OR REPLACE TABLE BIKESTORES.SALES.ORDER_ITEMS (
        order_id INT,
        item_id INT,
        product_id INT ,
        quantity INT ,
        list_price DECIMAL (10, 2) ,
        discount DECIMAL (4, 2) ,
    PRIMARY KEY (order_id, item_id)
        );
```

--**Loading data into snowflake is successful**

OUTPUT-;



---

2. Once the table has got created, there is a requirement of FOREIGN KEY implementation coming into picture where one needs to add (ALTER TABLE COMMAND) below foreign key on the table mentioned pointing to another table (READ ABOUT FOREIGN KEY) as:

--sales.staffs (store_id) -> sales.stores(store_id)

ALTER TABLE BIKESTORES.SALES.STAFFS ADD FOREIGN KEY (STORE_ID) REFERENCES BIKESTORES.SALES.STORES(STORE_ID);

--sales.staffs (manager_id) -> sales.staffs (staff_id)

ALTER TABLE BIKESTORES.SALES.STAFFS ADD FOREIGN KEY (manager_id) REFERENCES BIKESTORES.SALES.STAFFS (staff_id);

--production. Products (category_id) -> production.categories (category_id)

ALTER TABLE BIKESTORES.PRODUCTION.PRODUCTS ADD FOREIGN KEY (category_id) REFERENCES BIKESTORES.PRODUCTION.CATEGORIES(category_id);

--production.products(brand_id) -> production.brands (brand_id)

ALTER TABLE BIKESTORES.PRODUCTION.PRODUCTS ADD FOREIGN KEY (brand_id) REFERENCES BIKESTORES.PRODUCTION.BRANDS(brand_id);

--sales.orders (customer_id) -> sales.customers (customer_id)

ALTER TABLE BIKESTORES.SALES.ORDERS ADD FOREIGN KEY (customer_id) REFERENCES BIKESTORES.SALES.CUSTOMERS (customer_id);

--sales.orders(store_id) -> sales.stores (store_id)

ALTER TABLE BIKESTORES.sales.orders ADD FOREIGN KEY (store_id) REFERENCES BIKESTORES.sales.stores(store_id);

--sales.orders (staff_id) -> sales.staffs (staff_id)

ALTER TABLE BIKESTORES.sales.orders ADD FOREIGN KEY (staff_id) REFERENCES BIKESTORES.sales.staffs(staff_id);

--sales.order_items(order_id) -> sales.orders (order_id)

ALTER TABLE BIKESTORES.sales.order_items ADD FOREIGN KEY (order_id) REFERENCES BIKESTORES.sales.orders(order_id);

--sales.order_items (product_id) -> production.products (product_id)

ALTER TABLE BIKESTORES.sales.order_items ADD FOREIGN KEY (product_id) REFERENCES BIKESTORES.production.products(product_id);

--production.stocks (store_id) -> sales.stores (store_id)

ALTER TABLE BIKESTORES.production.stocks ADD FOREIGN KEY (store_id) REFERENCES BIKESTORES.sales.stores(store_id);

--production.stocks (product_id) -> production.products (product_id)

ALTER TABLE BIKESTORES.production.stocks ADD FOREIGN KEY (product_id) REFERENCES BIKESTORES.production.products(product_id);

3. Does any of the table has missing or NULL value ? If yes which are those and what are their counts ?

----THERE IS MISSING VALUE IN BIKESTORES.SALES.ORDERS COLOUMN (SHIPPED_DATE) ----

```
SELECT COUNT(*) AS missing_count          -- total= 170
FROM BIKESTORES.SALES.ORDERS
WHERE SHIPPED_DATE = '' ;
```

Results Chart

	...	MISSING_COUNT
1		170

---THERE IS MISSING VALUE IN BIKESTORES.SALES.CUSTOMERS CLOUMN (PHONE)----

```
SELECT COUNT(*) AS missing_count        -- total= 1267
FROM BIKESTORES.SALES.CUSTOMERS
WHERE PHONE = '' ;
```

Results Chart

	...	MISSING_COUNT
1		1267

4.Does the datasets has any DUPLICATE(identical rows) ? If yes – can you just keep the first record and remove all rest if its possible without using any JOINS or WINDOW function?

```
SELECT COUNT(*)
AS TOTL_ROWS FROM BIKESTORES.SALES.CUSTOMERS;  -- 1445
SELECT COUNT(DISTINCT FIRST_NAME, LAST_NAME, PHONE, EMAIL, STREET, CITY, STATE, ZIP_CODE) AS
TOTL_DISTINCT_ROWS
FROM BIKESTORES.SALES.CUSTOMERS;   --1445

SELECT COUNT(*)
AS TOTL_ROWS FROM BIKESTORES.SALES.ORDERS; -- 1615
SELECT COUNT(DISTINCT ORDER_ID, CUSTOMER_ID, ORDER_STATUS, ORDER_DATE, REQUIRED_DATE, SHIPPED_DATE,
STORE_ID,STAFF_ID) AS TOT_DISTINCT_ROWS
FROM SALES.ORDERS; -- 1615

SELECT COUNT(*) AS TOT_ROWS FROM BIKESTORES.SALES.ORDER_ITEMS; -- 4722
SELECT COUNT(DISTINCT ORDER_ID, ITEM_ID, PRODUCT_ID, QUANTITY, LIST_PRICE,DISCOUNT) AS TOT_DISTINCT_ROWS
FROM BIKESTORES.SALES.ORDER_ITEMS; -- 4722

SELECT COUNT(*) AS TOT_ROWS FROM BIKESTORES.SALES.STAFFS; -------10
SELECT COUNT(DISTINCT STAFF_ID, FIRST_NAME, LAST_NAME, EMAIL, PHONE, ACTIVE, STORE_ID, MANAGER_ID) AS
TOT_DISTINCT_ROWS
FROM SALES.STAFFS; -- 10
```

SELECT COUNT(*) AS TOT_ROWS FROM BIKESTORES.SALES.STORES; --
3

SELECT COUNT(DISTINCT STORE_NAME, PHONE, EMAIL, STREET, CITY,
STATE, ZIP_CODE) AS TOT_DISTINCT_ROWS

FROM SALES.STORES; -- 3

SELECT COUNT(*) AS TOT_ROWS FROM
BIKESTORES.PRODUCTION.BRANDS; -- 9

SELECT COUNT(DISTINCT BRAND_ID,BRAND_NAME) AS
TOT_DISTINCT_ROWS

FROM BIKESTORES.PRODUCTION.BRANDS; -- 9

SELECT COUNT(*) AS TOT_ROWS FROM PRODUCTION.CATEGORIES; -- 7

SELECT COUNT(DISTINCT CATEGORY_ID,CATEGORY_NAME) AS
TOT_DISTINCT_ROWS

FROM BIKESTORES.PRODUCTION.CATEGORIES; -- 7

SELECT COUNT(*) AS TOT_ROWS FROM
BIKESTORES.PRODUCTION.PRODUCTS; -- 321

SELECT COUNT(DISTINCT PRODUCT_ID, PRODUCT_NAME, BRAND_ID,
CATEGORY_ID, MODEL_YEAR, LIST_PRICE) AS

TOT_DISTINCT_ROWS

FROM BIKESTORES.PRODUCTION.PRODUCTS; -- 321

SELECT COUNT(*) AS TOT_ROWS FROM
BIKESTORES.PRODUCTION.STOCKS; -- 939

SELECT COUNT(DISTINCT STORE_ID, PRODUCT_ID, QUANTITY) AS
TOT_DISTINCT_ROWS

FROM BIKESTORES.PRODUCTION.STOCKS; -- 939

--**so there is no duplicate records and I am using distinct function to check whether there is duplicate records or not.....**

--

5.How many unique tables are present in each schema and under each table how many records are

we having ? (Write SQL Script for the same – I don't need answer like 3/5/4 etc)

---UNIQUE TABLES PRESENT IN EACH SCHEMAS----

DESC SCHEMA BIKESTORES.PRODUCTION;

Results | Chart

	created_on	name	kind
1	2023-12-05 21:28:04.896 -0800	BRANDS	TABLE
2	2023-12-06 00:10:09.431 -0800	CATEGORIES	TABLE
3	2023-12-05 21:28:59.035 -0800	PRODUCTS	TABLE
4	2023-12-05 21:29:45.120 -0800	STOCKS	TABLE

Query Details
Query duration 47ms
Rows 4
Query ID 01b0d84d-3200-f46a-...
created_on
100% filled

DESC SCHEMA BIKESTORES.SALES;

Results | Chart

	created_on	name	kind
1	2023-12-05 21:30:32.128 -0800	CUSTOMERS	TABLE
2	2023-12-06 00:05:20.464 -0800	ORDERS	TABLE
3	2023-12-05 21:32:03.148 -0800	ORDER_ITEMS	TABLE
4	2023-12-05 21:31:16.938 -0800	STAFFS	TABLE
5	2023-12-05 21:30:51.914 -0800	STORES	TABLE

Query Details
Query duration 27ms
Rows 5
Query ID 01b0d84e-3200-f4fe-0...
created_on
100% filled

---NUMBER OF RECORDS IN EACH TABLE---

```
SELECT
TABLE_SCHEMA,
TABLE_NAME,
ROW_COUNT
FROM INFORMATION_SCHEMA.TABLES
```

ORDER BY 1 DESC

limit 9;

Results Chart

	TABLE_SCHEMA	TABLE_NAME	ROW_COUNT
1	SALES	ORDERS	1615
2	SALES	STORES	3
3	SALES	STAFFS	10
4	SALES	ORDER_ITEMS	4722
5	SALES	CUSTOMERS	1445
6	PRODUCTION	BRANDS	9
7	PRODUCTION	STOCKS	939
8	PRODUCTION	CATEGORIES	7
9	PRODUCTION	PRODUCTS	321

Query Details

Query duration 2.6s

Rows 9

Query ID 01b0d851-3200-f4d5-0...

TABLE_SCHEMA A

SALES 5

PRODUCTION 4

TABLE_NAME A

100% filled

---

6. Which store has the highest number of sales ?

```
select store_id ,
count(order_id)as highest_num_of_sales
from BIKESTORES.SALES.ORDERS
group by 1
order by 2 desc;
```

Results Chart

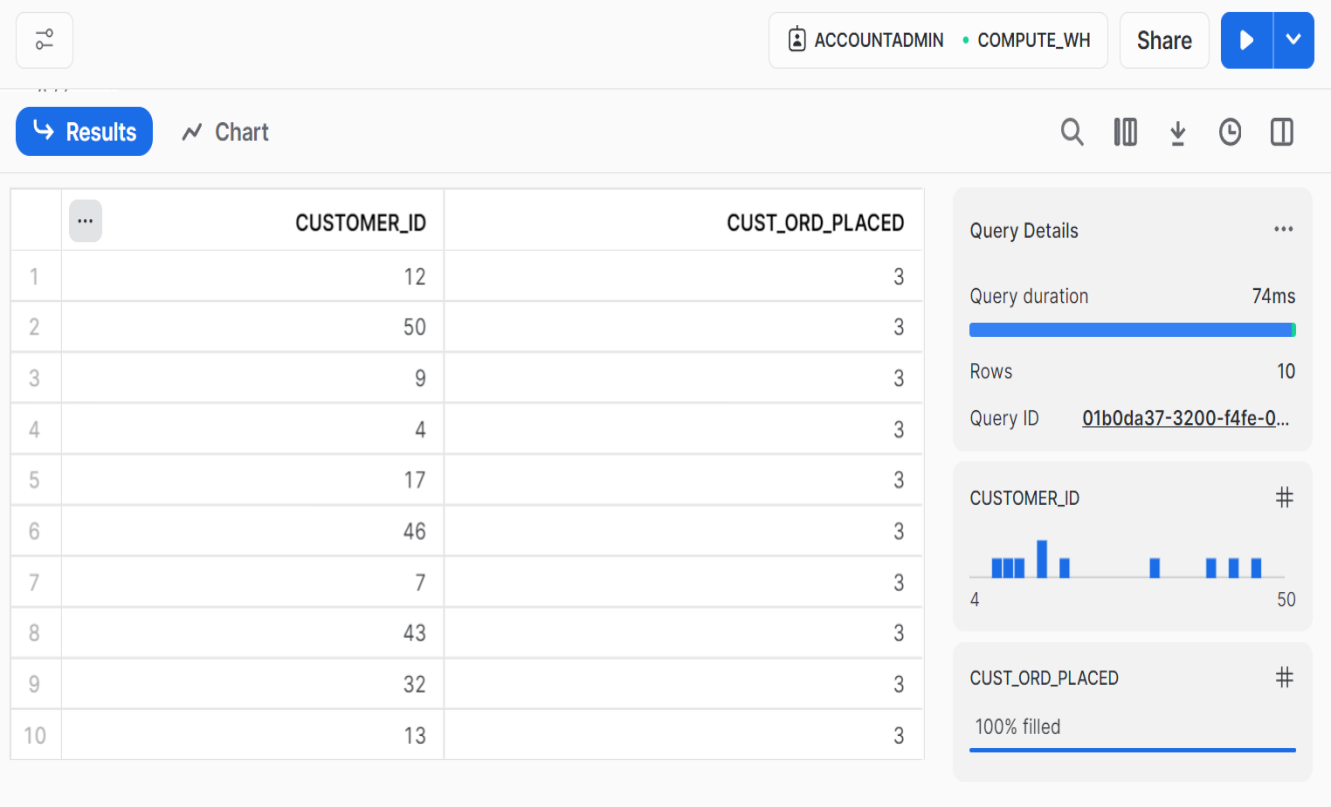
	STORE_ID	HIGHEST_NUM_OF_SALES
1	2	1093
2	1	348
3	3	174

---**store = 2 has highest number of sales.....**

---

8. How many orders each customer has placed (give me top 10 customers)

```
select CUSTOMER_ID,
count(order_id) as cust_ord_placed
from BIKESTORES.SALES.ORDERS
group by 1
order by 2 desc
limit 10;
```



	CUSTOMER_ID	CUST_ORD_PLACED
1	12	3
2	50	3
3	9	3
4	4	3
5	17	3
6	46	3
7	7	3
8	43	3
9	32	3
10	13	3

9. Which are the TOP 3 selling product ?

```
SELECT PRODUCT_ID,
ROUND(SUM(QUANTITY*LIST_PRICE,2) AS TOT_SALES
FROM BIKESTORES.SALES.ORDER_ITEMS
GROUP BY 1
ORDER BY 2 DESC
LIMIT 3;
```

Results Chart

	PRODUCT_ID	···	TOT_SALES
1	7		615998.46
2	9		434998.55
3	4		414698.57

10. Which was the first and last order placed by the customer who has placed maximum number of orders ?

```
SELECT CUSTOMER_ID,
MIN(ORDER_ID) AS First_Order,
MAX(ORDER_ID) as Last_Order
FROM BIKESTORES.SALES.ORDERS
GROUP BY 1
ORDER BY COUNT(ORDER_ID) DESC
LIMIT 1;
```

Results Chart

	··· CUSTOMER_ID	FIRST_ORDER	LAST_ORDER
1	31	162	1497

12. Which product has orders more than 200 ?

```
SELECT PRODUCT_ID, COUNT(DISTINCT ORDER_ID) AS TOT_ORDERS
FROM BIKESTORES.SALES.ORDER_ITEMS
GROUP BY 1
HAVING TOT_ORDERS > 200
ORDER BY 2 DESC;
```

Results Chart

	PRODUCT_ID	TOT_ORDERS
Query produced no results		

---- There is no product who has more than 200 orders..

13. Add a column TOTAL_PRICE with appropriate data type into the sales.order_items.

```
ALTER TABLE BIKESTORES.SALES.ORDER_ITEMS
ADD COLUMN TOTAL_PRICE DECIMAL(15,4);
```

Results Chart

	PRODUCT_ID	QUANTITY	LIST_PRICE	DISCOUNT	TOTAL_PRICE

14.Calculate TOTAL_PRICE = quantity * list price and Update the value for all rows in the sales.order_items table.

```
UPDATE BIKESTORES.SALES.ORDER_ITEMS
SET TOTAL_PRICE = ROUND(QUANTITY * LIST_PRICE,2);
```

TOTAL_PRICE
599.9900
3599.9800
3098.0000
1199.9800
2899.9900
599.9900

14.What is the value of the TOTAL_PRICE paid for all the sales.order_items ?

```
SELECT SUM(TOTAL_PRICE) AS TOT_PRICE_PAID FROM
BIKESTORES.SALES.ORDER_ITEMS;
```

Results Chart

	...	TOT_PRICE_PAID
1		8578988.8800

----------**THE END**----------