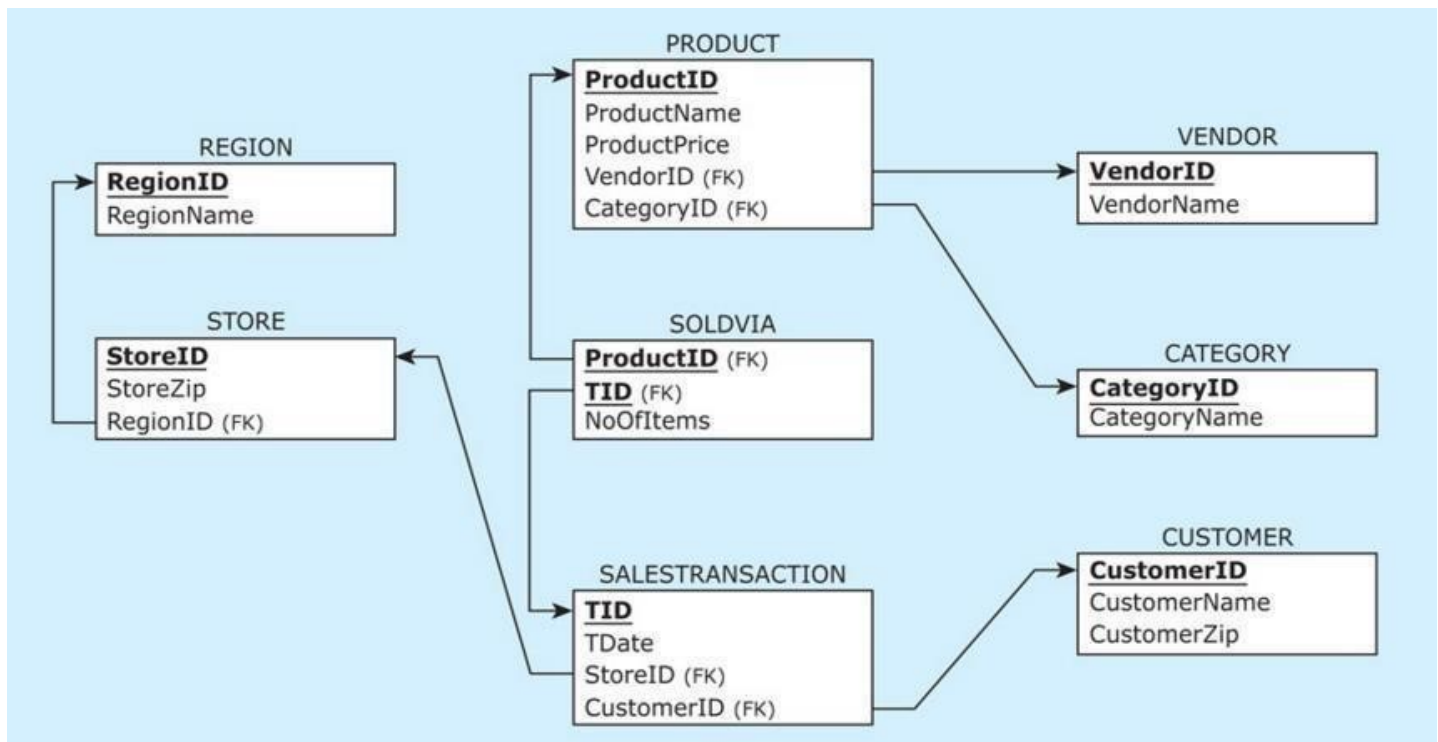


The aim of this lab is to get hands-on experience writing SQL queries. For this lab, we require two databases: same database of a retail company sales department and HAHF database. You can download the scripts for HAHF database from D2L.



SUBMISSION

Use Lab7_submission.sql file and copy your SQL statements into it and submit before the due time. Rename the file with your name, for example, as *Lab7_SaeedMirjalili.sql*.

GRADING

Each query is of 5 marks

QUERIES

1. Display the TID, CustomerName, and TDate of sales transactions that a customer bought a product whose ProductName is Dura Boot.
2. Display the ProductID and ProductName of the cheapest product.
3. Display the ProductID, ProductName, and VendorName for products whose price is below the average price of all products.
4. Display the Name of Customers and the average price of all products they bought.

5. Display Name of Customers who purchased the products which are more expensive (with price greater than) the average price of all products purchased by all customers.
6. Display the ProductID for the product that has been sold the most (i.e., that has been sold in the highest quantity).
7. Display the RegionID, RegionName, and number of stores in the region for all regions. Sort the results by number of stores from greatest to least.

VIEWS AND SET OPERATORS

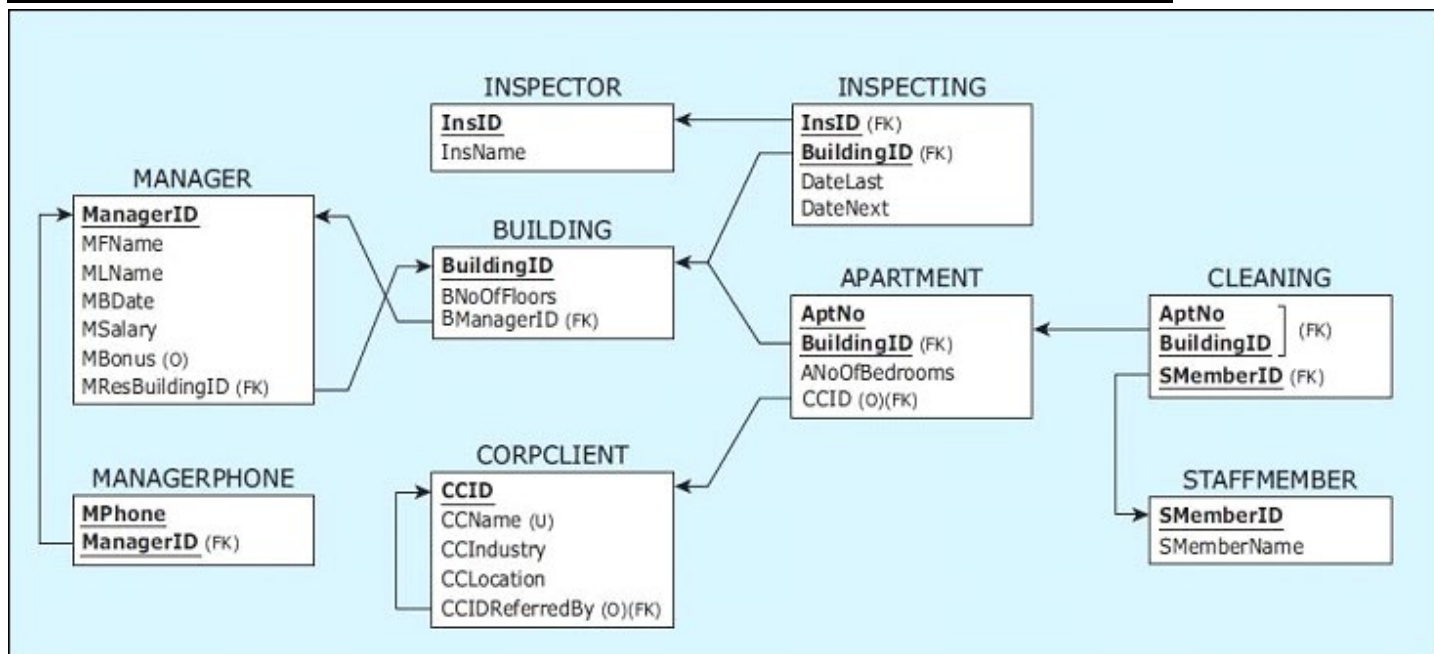
Create two view Product_More_Than_3_Sold and Products_in_Multiple_Trnsc as follows:

```
CREATE VIEW    Products_more_than_3_sold AS
SELECT        ProductId, ProductName, ProductPrice
FROM          Product
WHERE         ProductId IN (SELECT ProductId
                           FROM Soldvia
                           GROUP BY ProductId
                           HAVING SUM(noofitems) > 3
                           );
```

```
CREATE VIEW    Products_in_multiple_trnsc AS
SELECT        ProductId, ProductName, ProductPrice
FROM          Product
WHERE         ProductId IN (SELECT Productid
                           FROM SoldVia
                           GROUP BY ProductId
                           HAVING COUNT(*) > 1
                           );
```

Using these two views, write a query for the question 8, 9, and 10.

8. Retrieve the product ID, product name, and product price for each product that has more than three items sold within all sales transactions or whose items were sold in more than one sales transaction (Hint: UNION)
9. Retrieve the product ID, product name, and product price for each product that has more than three items sold within all sales transactions and whose items were sold in more than one sales transaction. (Hint : INTERSECTS)
10. Retrieve the product ID, product name, and product price for each product that has more than three items sold within all sales transactions but whose items were not sold in more than one sales transaction. (Hint: MINUS)

Relational schema: HAFH Realty Company Property Management Database

Download the create, insert, and drop scripts for above relational schema from BrightSpace-D2L. Make sure no table exist with same name as in script in your workspace.

Note that there are no values in the *MResBuildingID* in the relation *Manager*.

MANAGERID	MFNAME	MLNAME	MBDATE	MSALARY	MBONUS	MRESBUILDINGID
M12	Boris	Grant	01/04/1980	60000	-	-
M23	Austin	Lee	02/05/1975	50000	5000	-
M34	George	Sherman	12/06/1978	52000	2000	-

We will add the missing referential integrity constraint through ALTER statement.

Add the missing referential integrity constraint, in which the foreign key column *MResBuildingID* in the relation *Manager* refers to the primary key of the relation *Building* by issuing the following statement:

```

ALTER TABLE Manager
ADD CONSTRAINT fkresidesin
FOREIGN KEY (MResBuildingId) REFERENCES Building (BuildingId);
  
```

Once the referential integrity constraint *fkresidesin* is in place, you can add values to the *MResBuildingID* column of relation *Manager* by executing the following UPDATE statement:

```

UPDATE Manager
SET MResBuildingId = 'B1'
WHERE ManagerId = 'M12';
  
```

11. Update the manager table so that table look like below:

MANAGERID	MFNAME	MLNAME	MBDATE	MSALARY	MBONUS	MRESBUILDINGID
M12	Boris	Grant	01/04/1980	60000	-	B1
M23	Austin	Lee	02/05/1975	50000	5000	B2
M34	George	Sherman	12/06/1978	52000	2000	B4

12. Display the MFName, MLName, MSalary, MBDate, and number of buildings that the manager manages for all managers with a salary less than \$55,000.
13. Display the BuildingID and AptNo, for all apartments leased by the corporate client WindyCT.
14. Display the SMemberID and SMemberName of staff members cleaning apartments rented by corporate clients whose corporate location is Chicago. Do not display the same information more than once.
15. Display the CCName of the client and the CCName of the client who referred it, for every client referred by a client in the music industry.
16. Display the BuildingID, AptNo, and ANoOfBedrooms for all apartments that are not leased.