CPSC 2221

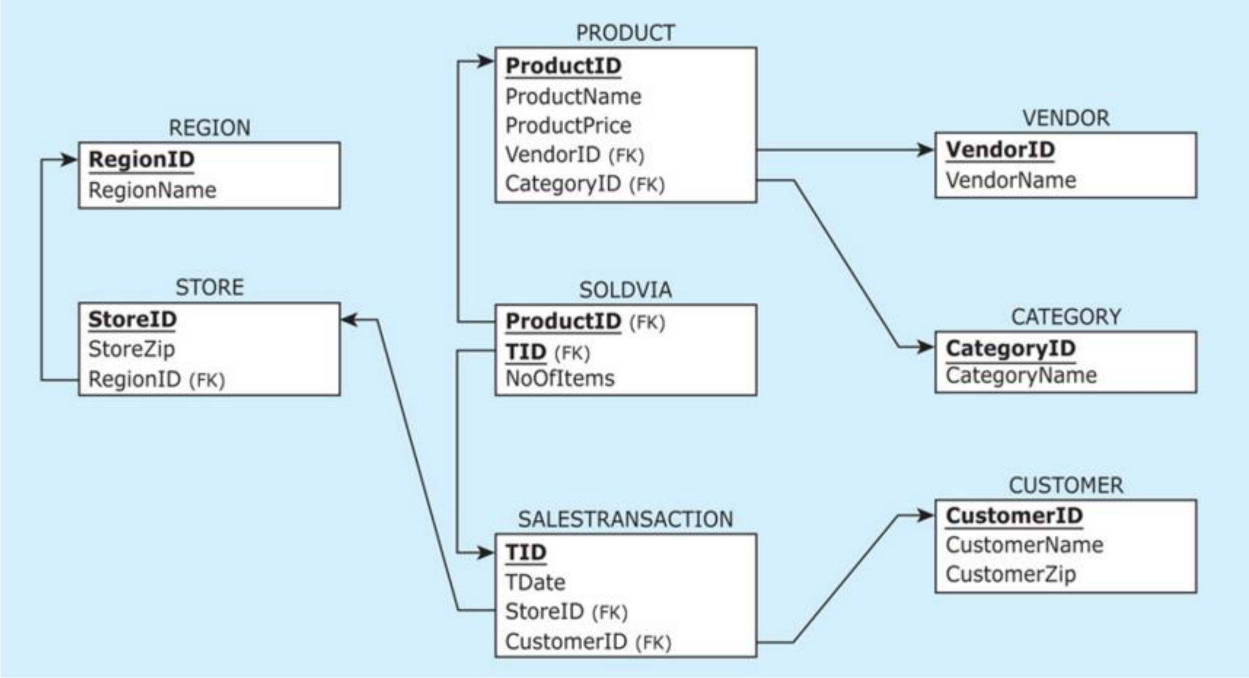
Lab Assignment 8: Advanced SQL

Total Marks: 100

Due Date: 11 Nov 2016

INDIVIDUAL LAB SUBMISSION

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



SUBMISSION

Submit a word document containing

✓ text query

✓ sql query and

✓ snapshot of query result.

Grading

➢ Each query is of 5 marks [SQL query and snapshot of query result]

QUERIES

1. Display the TID, CustomerName, and TDate for sales transactions involving a customer buying 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 ProductID for the product that has been sold the most (i.e., that has been sold in the highest quantity).

5. 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:

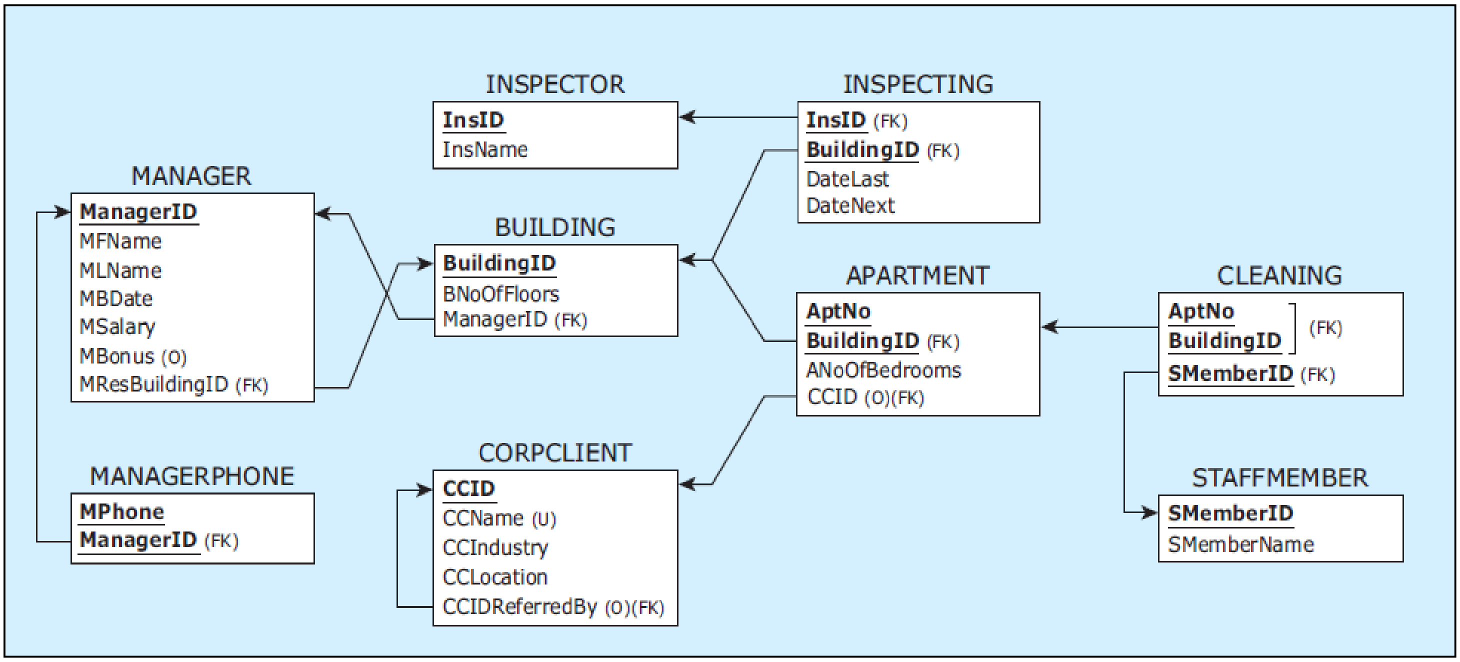
```
CREATE VIEW             products_more_than_3_sold AS
        SELECT          productid, productname, productprice
        FROM                    product
        WHERE                   productid IN
                        (SELECT         productid
                         FROM                   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);
```

6. 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)

7. 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)

8. 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 script for above relational schema from D2L. Make sure no table exist with same name as in script in your workspace.

9. Display the MFName, MLName, MSalary, MBDate, and number of buildings that the manager manages for all managers with a salary less than $55,000.

10. Display the BuildingID and AptNo, for all apartments leased by the corporate client WindyCT.

11. Display the InsID and InsName for all inspectors whose next inspection is scheduled after 1-JAN-2014. Do not display the same information more than once.

12. 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.

13. 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.

14. Display the BuildingID, AptNo, and ANoOfBedrooms for all apartments that are not leased.

15. Create a left outer join on the apartment and the corpclient to display building id, apartment number and client name.

16. Create a right outer join on the apartment and the corpclient to display building id, apartment number and client name.

17. Create a full outer join on the apartment and the corpclient to display building id, apartment number and client name.

18. Retrieve records for all buildings that do not have managers living in them.

19. Display the BuildingID, BNoOfFloors, and the manager's MFName and MLName for all buildings.

20. Create **ONE** trigger on any table and show the results when trigger gets activated. (Hint: Create a trigger based on the count of the tuples, Updating salary value of the manager etc.). Below is an example of a trigger based on count of tuples:

```
create or replace TRIGGER apartmentinserttrigger
 BEFORE INSERT   ON apartment referencing NEW AS newrow
 FOR EACH row
 DECLARE row_count number;
 BEGIN
 SELECT count(*) INTO   row_count   FROM   apartment where ccid = :newrow.ccid;
 IF (row_count >= 5) then
  :newrow.ccid := NULL;
  END IF;
  END;
```