FIT5137 Group Assignment - Sem 2/2017 (Weight = 30%)
Due date: Week 12, Monday 16-Oct-2017, 11:55pm

A. General Information and Submission

- This is a group assignment. One group consists of 2-3 students only. You need to report your group composition through the link below as soon as possible. This Google Form should only be filled in by one member of the group. https://goo.gl/forms/odf5FbVxcltlQIDY2
- *Submission method*: Submission is online through Moodle. Only one group member needs to upload the assignment to Moodle
- *Penalty for late submission*: 10% deduction for each day
- *Oracle account details*: You will need to supply with this assignment an Oracle username and password, used for this assignment.
- *Assignment Coversheet*: You will need to sign the assignment coversheet

B. Problem Description – Human Resources & Sales Order Warehouse

Human Resources & Sales Order Database records the information about employees and facilities, and tracks product inventories and sales of company's products through various sales channels. Our database stores order transactions during the period 01-Jan-2004 to 31-Dec-2008. The database tables can be found at **hosales**. You can, for example, execute the following query:

```
select * from hosales.<table_name>;
```

The data definition of each table in **hosales** is as follows:

Table Name (PK/FK)	Attributes and Data Types		Notes
Employees PRIMARY KEY: Employee_ID FOREIGN KEYS: Manager_ID Department_ID Job_ID	Employee_ID	NUMERIC	This table stores employees information. Each employee has an identifying ID, e-mail address, job identifying ID, salary (per month), and manager. Some employees earn commissions in addition to their salary.
	First_Name	VARCHAR	
	Last_Name	VARCHAR	
	Email	VARCHAR	
	Phone_Number	VARCHAR	
	Hire_Date	DATE	
	Job_ID	VARCHAR	
	Salary	NUMERIC	
	Commission_Pct	NUMERIC	
	Manager_ID	NUMERIC	
	Department_ID	NUMERIC	
Departments PRIMARY KEY: Department_ID FOREIGN KEYS: Manager_ID Location_ID	Department_ID	NUMERIC	This table stores departments information.
	Department_Name	VARCHAR	
	Manager_ID	NUMERIC	
	Location_ID	NUMERIC	

Table Name (PK/FK)	Attributes and Data Types		Notes
Jobs PRIMARY KEY: Job_ID	Job_ID	VARCHAR	This table stores jobs information. Each job has an identifying ID, job title, and a minimum and maximum salary range for the job.
	Job_Title	VARCHAR	
	Min_Salary	NUMERIC	
	Max_Salary	NUMERIC	
Job_History PRIMARY KEY: Employee_ID Start_Date Job_ID FOREIGN KEYS: Job_ID Employee_ID	Employee_ID	NUMERIC	This table stores job history information. Some employees have been with the company for a long time and have held different positions within the company. When an employee resigns, the duration the employee was working, the job identifying ID, and the department are recorded.
	Start_Date	DATE	
	End_Date	DATE	
	Job_ID	VARCHAR	
	Department_ID	NUMERIC	
Regions PRIMARY KEY: Region_ID	Region_ID	NUMERIC	This table stores regions information.
	Region_Name	VARCHAR	
Countries PRIMARY KEY: Country_ID FOREIGN KEYS: Region_ID	Country_ID	CHAR	This table stores countries information.
	Country_Name	VARCHAR	
	Region_ID	NUMERIC	
Locations PRIMARY KEY: Location_ID FOREIGN KEYS: Country_ID	Location_ID	NUMERIC	This table stores locations information.
	Street_Address	VARCHAR	
	Postal_Code	VARCHAR	
	City	VARCHAR	
	State_Province	VARCHAR	
	Country_ID	CHAR	

Table Name (PK/FK)	Attributes and Data Types		Notes
Customers PRIMARY KEY: Customer_ID FOREIGN KEYS: Country_ID Account_Mgr_ID	Customer_ID	NUMERIC	This table stores customers information. Each customer has an identifying ID. Customer information include customer name, street name, city or province, country, phone numbers, and postal code. Some customers place orders through the Internet, so e-mail addresses are also recorded. Because of language differences among customers, the company records the native language and territory of each customer. The company places a credit limit on its customers, to limit the amount of products they can purchase at one time. Some customers have an account manager, and this information is also recorded. * Customer credit type: low credit (credit <=1500), medium credit (1500<credit <=3500), high credit (credit > 3500)
	Cust_first_name	VARCHAR	
	Cust_last_name	VARCHAR	
	Cust_address	VARCHAR	
	Postcode	NUMERIC	
	City	VARCHAR	
	State	VARCHAR	
	Country_ID	CHAR	
	Phone_Numbers	VARCHAR	
	Nls_Language	VARCHAR	
	Nls_Territory	VARCHAR	
	Credit_Limit	NUMERIC	
	Cust_Email	VARCHAR	
	Account_Mgr_ID	NUMERIC	
Products PRIMARY KEY: Product_id	Product_ID	NUMERIC	This table stores products information. The company sells several products, such as computer hardware and software, and tools. The company maintains information about these products, such as product identifying ID, the category, the weight group (for shipping purposes), the supplier, the availability status of the product, a list price, a minimum price at which a product will be sold, and a URL address for manufacturer information.
	Product_Name	VARCHAR	
	Product_Description	VARCHAR	
	Category_ID	NUMERIC	
	Weight_Class	NUMERIC	
	Supplier_ID	NUMERIC	
	Product_Status	VARCHAR	
	List_Price	NUMERIC	
	Min_Price	NUMERIC	
	Catalog_URL	VARCHAR	

Table Name (PK/FK)	Attributes and Data Types		Notes
Orders PRIMARY KEY: Order_ID FOREIGN KEYS: Customer_ID Sales_Rep_ID Promotion_ID	Order_ID	NUMERIC	This table stores orders information. When a customer places an order, the company tracks the date of the order, how the order was placed, the current status of the order, total amount of the order, and the sales representative who helped place the order. The sales representative may or may not be the same person as the account manager for a customer. If an order is placed over the Internet, no sales representative is recorded.
	Order_Date	TIMESTAMP	
	Order_Mode	VARCHAR	
	Customer_ID	NUMERIC	
	Order_Status	NUMERIC	
	Order_Total	NUMERIC	
	Sales_Rep_ID	NUMERIC	
	Promotion_ID	NUMERIC	
	Total_Price	NUMERIC	
Order_Items PRIMARY KEY: Order_ID Line_Item_ID FOREIGN KEYS: Product_ID	Order_ID	NUMERIC	This table stores orders information. The company tracks the number of items ordered, the unit price, and the products ordered.
	Line_Item_ID	NUMERIC	
	Product_ID	NUMERIC	
	Unit_Price	NUMERIC	
	Quantity	NUMERIC	
Warehouses PRIMARY KEY: Warehouse_ID FOREIGN KEYS: Location_ID	Warehouse_ID	NUMERIC	This table stores warehouses information. Each warehouse has a warehouse identifying ID, name, facility description, and location identification number.
	Warehouse_Spec	VARCHAR	
	Warehouse_Name	VARCHAR	
	Location_ID	NUMERIC	
Inventories PRIMARY KEY: Product_ID Warehouse_ID FOREIGN KEYS: Product_ID Warehouse_ID	Product_ID	NUMERIC	This table stores inventories information.
	Warehouse_ID	NUMERIC	

Table Name (PK/FK)	Attributes and Data Types		Notes
Promotions PRIMARY KEY: Promotion_ID	Promotion_ID	NUMERIC	This table stores promotions information.
	Discount	VARCHAR	
	Pro_Desc	VARCHAR	
	Start_Date	DATE	
	End_Date	DATE	

C. Tasks

The assignment is divided into **FIVE (5)** main tasks:

1. Design a data warehouse for the above HOSALES database. The output of this task is an ER diagram and proper data cleaning processes.

   Before you design the data warehouse, you need to make sure that you have explored the operational database, and have done data cleaning when necessary. If you have done the data cleaning process, you need to explain what strategies you have taken to explore and clean the data.

   The outputs of this task are:

   (a) The **E/R diagram** of the operational database,

   (b) If you have done the data cleaning, explain what kind of data cleaning process that you have done (you need to show the SQL to detect the data cleaning errors in the operational database, and SQL of the data cleaning, as well as the **screenshot** of data *before* and *after* data cleaning)

2. Draw multi-fact star schemas

   The star schema for this data warehouse will have **three fact tables**: one is related to the **Employee**, another is related to the **Customer**, and the last one is related to **Sales Order**. You need to identify the fact measures and dimensions.

   The following queries might help you to identify the fact measures and dimensions:

   - How many online orders in summer?
   - What is the total salary by each department and each job in each month?
   - What is the number of employees by each country and by each city?
   - What is the average sales from America in 2007?
   - What is the total number of high credit customers from the US?
   - What is the total sales by each product and warehouses in each year/month?

Besides the above 6 queries, you need to come up with **TWO (2) additional queries** which you use to identify the fact measures and dimensions. Each query at least contains two dimensions and a fact measurement, and should be useful to the management.

You need to do **TWO VERSIONS** of star schema. The two versions of the star schemas must feature: **Hierarchy**, **Bridge Tables** (with Weight and ListAgg), and **Temporal**. In each version, you need to specify where the Hierarchy, Bridge Tables, and Temporal are. You need to list the name of the fact table using the following format.

	Version-1	**Version-2**
Hierarchy	Fact_Table_Name	
Bridge Table		Fact_Table_Name
Temporal	Fact_Table_Name (SCD Type)	Fact_Table_Name (SCD Type)

Figure 2.1 An example of two versions of the star schemas

You need to **compare version-1 and version-2,** by using Hierarchy vs. Non-Hierarchy, or using different temporal technique (e.g. Temporal data warehousing SCD Type 4 vs. SCD Type 3), and by using a Bridge Table.

For example: in Version-1, the Employee fact table uses a Hierarchy approach, whereas in Version-2, the Employee fact table uses a Non-Hierarchy approach (so you are comparing Hierarchy vs. Non-Hierarchy using Employee fact table between Version-1 and Version-2)

The other possibility is, for example, in Version-1, the Employee fact table uses Temporal data warehousing SCD Type 4, but Version-2 uses SCD Type 2 (so in this case, you are comparing SCD Type 4 and SCD Type 2 using the Employee fact table).

*So, for each star schema, you need to pick one or two features that you would like to compare. But overall in your two versions of the star schemas, you must use **Hierarchy**, **Bridge Table**, and **Temporal**. You can discuss with your tutor, which options you would like to choose for your version-1 and version-2.*

The outputs of this task are:

(a) Two queries that you come up by yourselves,

(b) Two versions of star schema diagrams and a table like Figure 2.1, and

(c) A short explanation for each of the star schemas.

3. Implement the **two** (2) star schemas using **SQL**

   You need to implement the star schemas that you have drawn in Task 2 above. It means that you need to create the fact and dimension tables, and populate these tables.

   When naming the fact tables and dimension tables, you need to give the identical name for the two versions and end with the version number to differentiate them. For example, "Fact_sales_v1" for version-1 and "Fact_sales_v2" for version-2.

   The output is a series of SQL statements to perform this task. You will also need to show that this task has been carried out successfully.

   You also need to compare between the two versions that you have created, showing the tables, and explaining the differences and the impact (to the design, to the storage, and to the query processing) from these differences.

   If your account is full, you will need to drop all of the tables that you have previously created during the tutorials.

   The outputs of this task are:

   (a) SQL statements (e.g. create table, insert into, etc) to create the star schemas Version-1

   (b) SQL statements (e.g. create table, insert into, etc) to create the star schemas Version-2

   (c) Screen shots of the tables that you have created; this includes the contents of each table that you have created. If the table is very big, you can print a snapshot of the contents of the table.

   (d) A comparison between the two versions of the star schemas

4. Create the following reports using OLAP queries (in total there are **eight** reports). You need to use **the star schema version-1** to generate the following reports.

   a. *Reports with proper sub-totals: REPORT 1 and REPORT 2*

      Produce **two** reports. Choose any two queries from the above six listed queries.

      These new reports include sub-totals, using the Cube or Roll-up or Partial Cube/Roll-up operators. You need to use Decode and Grouping to generate the reports with good format.

The outputs of this task are:

(*a*) The query questions written in English,

(*b*) Your explanation on why such a query is necessary or useful for the management,

(*c*) The SQL commands that include sub-totals, using the Cube or Roll-up or Partial Cube/Roll-up operators, and

(*d*) The screenshots of the query results (or part of the query results).

b. *Reports with Rank and Percent_Rank: REPORT 3 and REPORT 4*

Produce **two** reports that contain rank and percent_rank using the queries that are come up by yourself.

The outputs of this task are:

(*a*) The query questions written in English,

(*b*) Your explanation on why such a query is necessary or useful for the management,

(*c*) The SQL commands that contains rank and percent_rank, and

(*d*) The screenshots of the query results (or part of the query results).

c. *Reports with Partitions: REPORT 5 and REPORT 6*

Produce **two** reports that contain partitions (which should be different from any of the reports above) that are useful for management. Each query should cover two different dimensions and one fact measurement.

The outputs of this task are:

(*a*) The query questions written in English,

(*b*) Your explanation on why such a query is necessary or useful for the management,

(*c*) The SQL commands that contains partitions, and

(*d*) The screenshots of the query results (or part of the query results).

d. *Reports with moving and cumulative aggregates: REPORT 7 and REPORT 8*

Produce **two** reports containing moving and cumulative aggregates (which should be different from any of the reports above) that are useful for management. Each query should cover two different dimensions and one fact measurement.

The outputs of this task are:

(*a*) The query questions written in English,

(*b*) Your explanation on why such a query is necessary or useful for the management,

(*c*) The SQL commands that contains moving and cumulative aggregates, and

(*d*) The screenshots of the query results (or part of the query results).

5. For each of the queries in Task 4 above, you will need to:
   a. Show the access plans (and query trees), and the execution times,
   b. Create another set of queries, which will do the same job as Task 4, but will have different access plans (and query trees) and execution time. You will need to show that these queries produce the same results as those in Task 4. **You must use a variety of Hints and query optimization methods**; and
   c. For each pair of queries, that is two queries producing the same results but with different access plan/execution time, you need to discuss why one query is better than the other.

The format of the output of this task is as follows: For each query, you need to have

(*a*) The SQL from Task 4 above,

(*b*) The screenshot of the query result (from Task 4 above),

(*c*) The **execution plan** of the original query from Task 4 above,

(*d*) The **query tree** of the original query from Task 4 above,

(*e*) The new SQL (see Task 5b above),

(*f*) The screenshot of the query result from Task 5b,

(*g*) The **execution plan** of the new query from Task 5b,

(*h*) The **query tree** of the new query from Task 5b, and

(*i*) An explanation on why one query is better than the other.

D. Submission Checklist

1. One **combined pdf file** containing all tasks mentioned above:
   - ☐ Cover page
   - ☐ A signed coversheet
   - ☐ Details of your ORACLE accounts
   - ☐ A contribution declaration form:

     Each student must state the parts of the assignment that he/she did, otherwise your assignment will not be marked. An example is as follows:

     Percentage of contribution:
     1. Name: Adam, ID: 210008, Contribution: 40%
     2. Name: Ben, ID: 230933, Contribution: 30%
     3. Name: Chris, ID: 240934, Contribution: 30%

     List of parts that each student did:
     1. Adam: list the parts that Adam did
     2. Ben:
     3. Chris

   - ☐ Task C.1 (outputs *a, b*)
   - ☐ Task C.2 Draw Multiple Schemas (outputs *a, b, c*)
   - ☐ Task C.3 SQL Implementation of the Star Schemas and the comparison (outputs *a, b, c, d*)
   - ☐ Task C.4 Reports with Proper sub-totals (outputs *a, b, c, d*)
   - ☐ Task C.4 Reports with Rank and Percent_Rank (outputs *a, b, c, d*)
   - ☐ Task C.4 Reports with Partitions (outputs *a, b, c, d*)
   - ☐ Task C.4 Reports with Moving and Cumulative Aggregates (outputs *a, b, c, d*)
   - ☐ Task C.5 (outputs *a, b, c, d, e, f, g, h, i*)

2. **txt files** from the following tasks:
   - ☐ Task C.1 (SQL command as required by output *b*)
   - ☐ Task C.3 Implement Star Schemas (SQL command as required by output *a and b*)
   - ☐ Task C.4 Reports with Proper sub-totals (SQL command as required by output *c*)
   - ☐ Task C.4 Reports with Rank and Percent_rank (SQL command as required by output *c*)
   - ☐ Task C.4 Reports with Partitions (SQL command as required by output *c*)

- ☐ Task C.4 Reports with Moving and Cumulative Aggregates (SQL command as required by output *c*)
- ☐ Task C.5 (SQL commands as required by outputs *a, e*)

All of the above txt files must be run-able in Oracle.

3. Zip all the files above (pdf from #1 above, and txt files from #2 above), and upload this zip file to Moodle.

THE END