## Assignment 2

## Database Implementation and Queries

## Due Date:

Friday Week 11 (See Course Description for further date and time).

## Objectives:

To analyse and comprehend a provided ER diagram and Database Schema

To implement a database based on the provided ER diagram and Database Schema

To write required SQL statements to query the database

## Project Specification

The management team now require a partial implementation of the design made in Assignment 1. In order to keep consistency between the assignments, database specification containing the ER diagram and the schema is provided in this document. You should create your database according to this documentation. Please make sure that your implementation is consistent with this design. This means that your table names (upper case), field names (mixed case, no spaces) and data types have to be according to the specifications provided in this document. The implementation phase includes writing SQL statements to create a database and its tables, populating the tables with data, writing a number of queries to create reports that can be used by the management team. Your database should contain sufficient data in each table (5 – 10 records in each table) to demonstrate that your queries work.

## Implementation of the Database

To implement the database system, you are required to

1. provide SQL commands to create the database, its tables, the relationships of the tables
2. Create a text file called YourStudentId-Create.sql (format xxxxxxx-Create.sql) for example 2225991-Create.sql that will provide SQL commands to:
3. create a database called WareMart[YourStudentID] (eg WareMart30011111)
4. create all of the required tables for the database including their primary keys, foreign keys and the relationships of tables.
5. Create a text file called YourStudentId-Insert.sql (format xxxxxxx-Insert.sql) for example 2225991-Insert.sql that will provide SQL commands to:
6. insert **sufficient** data into each table you have created to test the queries (at least 5 records in each table).
7. You **are required** to include our full name as one of the clients; but you can provide fake details for your address.
8. Create a text file called YourStudentId-Queries.sql (format xxxxxxx-Queries.sql - for example 2225991-Queries.sql) that contains all of the queries to display the following reports:
9. An alphabetically sorted list of all clients. Only client number and name are required.
10. List of names and complete address of all employees sorted by their salary.
11. The date on which the most recent stock request has been made. The date itself will suffice.
12. List of all the client names and their residential addresses.
13. A list of all clients that have not placed a stock request yet. Displaying client number will be sufficient.
14. A list containing the name (surname and first name) of any employee that has picked any product(s) for a stock request.
15. A list containing the total quantity on hand for each product (product no and description) regardless of warehouses and location.
16. A list showing each product requested on each client stock request. Show client name, product number and quantity requested. sorted by client name and then product number.
17. A list of employees (surname and first name are sufficient) and their salary for all employees whose salary is less than or equal to average salary.
18. A list of employees as in question 9, but show their salary with a 7.5% increase.

|  |
| --- |
| **ITECH5006 Students Only** |
| 1. A list of all products (product number and description) and the quantity on hand for that product for each location at which it is stored within each warehouse. Sort it by product number and then place all locations (warehouse number and location number) for a given product together. 2. A list showing product number, the quantity requested, the quantity picked and the difference between the two. For products stored in more than one location within a warehouse the quantities should be added together. 3. A list of supervisors (staffid, surname and first name) and all of their subordinates (staffid, surname and first name). |

## Note: There are some general requirements when defining your select queries:

**You are required to adhere to the following output formatting conventions:**

* Any query requiring *names of people* should be printed as GivenName FamilyName (e.g. John Smith) in a column labelled NAME
* Any query requiring *addresses* should be printed as Street, Suburb State Postcode (e.g. 123 Anzac Pde, Maroubra NSW 2038) in a column labelled ADDRESS
* All *monetary* values should be printed with a dollar symbol ($), two digits after the decimal point, and with space for 7 digits before the decimal point
* You must use consistent and legible formatting in laying out your SQL queries. Include (brief) comments for any query or procedure that uses an "unusual" approach.

## What to submit

An electronic copy of your assignment should be submitted through Moodle and should include a copy of your report, completed according to the University of Ballarat Guide for the Presentation of Academic Work (available http://www.ballarat.edu.au/generalguide) and the **three** files described in *Create, insert and query statements* above.

Your document should include:

* A copy of the SITE Assignment Coversheet that includes a copy of the plagiarism statement.
* A copy of the SQL statements required to:
  + create the database and tables;
  + insert sufficient sample data into each table to demonstrate that your queries work.
  + a report of the results from running the SQL queries by using Copy/paste of their output.
* A bibliography containing a list of all resources used to complete the assignment. If no resources, apart from the course materials, have been used please indicate this.

## Assessment Criteria

* How clear and well organised your presentation is. On the front page of your report you should include a list of acknowledgements of all people who have assisted you with this assignment including fellow students, along with a statement of completion.
* Adherence to our standards. How clear and well organised your presentation is. You should write all the queries in consistent style and *use* *indent* *format*.
* Data correctness and quality. Please use appropriate data for your examples (e.g. do not use inappropriate person names)
* Joining of data from multiple tables should be completed using a WHERE statement only. JOINs are not to be used within any of the SQL statements. Use of any JOINs will result in **0 (zero)** marks being allocated for each SQL statement that utilizes them.
* Please refer to the provided marking guide (below) to see the distribution of marks.

## Assignment Resources:

* The Standard ER Diagram
* Relational Database Schema

*Hint: you need to decide the order that tables need to be created; and the order of tables in which data need to be inserted into.*

# WAREHOUSE

warehouseID

street

city

state

postcode

*managerID*

# EMPLOYEE

staffID

surname

firstName

dob

street

city

state

postcode

salary

*warehouseID*

*supervisedBy*

# LOCATION

warehouseID

locationID

aisle

shelf

bin

capacity

hazardousAllowed

# PRODUCT

productNum

description

packSize

*dptNumber*

# PROD\_LOCATION

warehouseID

locationID

productNum

quantityOnHand

# DEPARTMENT

dptNumber

dptName

# CLIENT

clientNum

clientName

# CLIENT\_ADDRESS

clientNum

addressType

street

city

state

postcode

# STOCK\_REQUEST

requestNum

requestDate

*clientNum*

# PICKING\_LIST

warehouseID

locationID

productNum

requestNum

quantityPicked

datePicked

*pickerStaffID*

# REQUEST\_LIST

requestNum

productNum

qtyRequested

Manages

Managed by

Supervised by

Supervises

Contains

Recruits

Assigned

Contains

Made by

Makes

Contains

Belongs to

Selects items

from

Has

Belongs to

Stored at

Associated with

Has

Belongs to

Works at

Appears on

## 2. Relational Database Schema

**WAREHOUSE**

|  |  |  |
| --- | --- | --- |
| **Column name** | **Data type** | **Comments** |
| warehouseID | VARCHAR(3) | Primary key |
| street | VARCHAR(20) |  |
| city | VARCHAR(15) |  |
| state | VARCHAR(3) | Examples – VIC, NSW, QLD |
| postcode | VARCHAR(4) | Examples – 3350, 2001, 3001 |
| managerID | INT | FK – References MPLOYEE.StaffID |

**EMPLOYEE**

|  |  |  |
| --- | --- | --- |
| **Column name** | **Data type** | **Comments** |
| staffID | INT A\_I | Primary key |
| surname | VARCHAR(20) |  |
| firstName | VARCHAR(15) |  |
| dob | Date | Short date – Example 06/09/1982 |
| street | VARCHAR(20) |  |
| city | VARCHAR(15) |  |
| state | VARCHAR(3) | Examples – VIC, NSW, QLD, TAS |
| postcode | VARCHAR(4) | Examples – 3350, 3355, 2001, 3001 |
| salary | Decimal(19,4) |  |
| warehouseID | VARCHAR(3) | FK – References AREHOUSE. warehouseID |
| supervisedBy | INT | FK – References MPLOYEE.StaffID |

**LOCATION**

|  |  |  |
| --- | --- | --- |
| **Column name** | **Data type** | **Comments** |
| warehouseID | VARCHAR(3) | Primary key - Examples W01, W02  FK – References WAREHOUSE. warehouseID |
| locationID | VARCHAR(3) | Primary key -Examples L01, L02 |
| Aisle | INT |  |
| Shelf | INT |  |
| Bin | INT |  |
| capacity | Double | Capacity in cubic meters |

**PROD\_LOCATION**

|  |  |  |
| --- | --- | --- |
| **Column name** | **Data type** | **Comments** |
| warehouseID | VARCHAR(3) | Primary key  FK - References WAREHOUSE. warehouseID |
| locationID | VARCHAR(3) | Primary key  FK – References LOCATION. locationID |
| productNum | INT | Primary key  FK – References PRODUCT. productNum |
| quantityOnHand | INT |  |

**PRODUCT**

|  |  |  |
| --- | --- | --- |
| **Column name** | **Data type** | **Comments** |
| productNum | INT A\_I | Primary key |
| description | VARCHAR(30) |  |
| packSize | INT |  |
| Price  dptNumber | Decimal (10,2)  INT | FK – References DEPARTMENT. dptNumber |

**DEPARTMENT**

|  |  |  |
| --- | --- | --- |
| **Column name** | **Data type** | **Comments** |
| dptNumber | INT A\_I | Primary key |
| dptName | VARCHAR(20) |  |

**CLIENT**

|  |  |  |
| --- | --- | --- |
| **Column name** | **Data type** | **Comments** |
| clientNum | INT A\_I | Primary key |
| clientName | VARCHAR(40) |  |

**CLIENT\_ADDRESS**

|  |  |  |
| --- | --- | --- |
| **Column name** | **Data type** | **Comments** |
| clientNum | INT A\_I | Primary key  FK – References CLIENT.clientNum |
| addressType | VARCHAR(1) | Primary key. Expected examples ‘R’ for residential, ‘M’ for main and ‘P’ for postal |
| street | VARCHAR(20) |  |
| city | VARCHAR(15) |  |
| state | VARCHAR(3) | Examples – VIC, NSW, QLD |
| postcode | VARCHAR(4) | Examples – 3350, 2001, 3001 |

**STOCK\_REQUEST**

|  |  |  |
| --- | --- | --- |
| **Column name** | **Data type** | **Comments** |
| requestNum | INT A\_I | Primary key |
| requestDate | Date | Short date – Example 06/09/2012 |
| clientNum | INT | FK – References CLIENT.clientNum |

**REQUEST\_LIST**

|  |  |  |
| --- | --- | --- |
| **Column name** | **Data type** | **Comments** |
| requestNum | INT | Primary key  FK – References STOCK\_REQUEST.requestNum |
| productNum | INT | Primary key  FK – References  PRODUCT.productNum |
| qtyRequested | INT |  |

**PICKING\_LIST**

|  |  |  |
| --- | --- | --- |
| **Column name** | **Data type** | **Comments** |
| warehouseID | VARCHAR(3) | Primary key.  FK – References WAREHOUSE. warehouseID |
| locationID | VARCHAR(3) | Primary key.  FK – References LOCATION. locationID |
| productNum | INT | Primary key.  FK – References PRODUCT. productNum |
| requestNum | INT | Primary key.  FK – References STOCK\_REQUEST.requestNum |
| quantityPicked | INT |  |
| datePicked | Date | Short date – Example 06/09/2014 |
| pickerStaffID | INT | Foreign Key – References EMPLOYEE.staffID |

End of document…

## Assignment 2 – Marking Overview

|  |  |
| --- | --- |
| **Documentation** | **/5** |
| **Create statements** | **/15** |
| **insert statements** | **/15** |
| **Queries** | **/45** |
| **Formatting** | **/10** |
| **ITECH5006 Queries** | **/15** |
|  |  |
| Total: | **ITECH1006**  **/90**  **ITECH5006**  **/105** |

**ITECH5006 Assignment 2 Marking Guide**

Student name: Student ID: Marker: Faezeh Afshar

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **ITEM** |  | **Max** | **Mark** |
| **Documentation** | | | **/5** |  |
|  |  | document well laid out and easy to read  contains a coversheet or front page  contains a table of contents and is set out according to the university guidelines  contains all of the required components of the assignment  bibliography included in APA format | 1  1  1  1  1 |  |
| **Database Design** | | | **/10** |  |
|  | **Revised Schema** | Includes (all/most/only minor/no) tables and attributes  Primary keys and foreign keys identified (including the table and key that the key related to)  Used appropriate type (i.e VARCHAR instead of TEXT, Postcodes are included as VarChar (4) ie not a number), Includes appropriate size/length of strings.  and clearly presented formatting of fields included eg dates | 3  4  2  1 |  |
| Create and insert statement | | | **/20** |  |
|  | **Create Statements** | Create (all/most/some/no) of the relevant new tables: Database correctly named and created (Includes own name and number), includes all of the PKs and FKs in the database. No evidence that tables have been created using the GUI. They work properly | 10 |  |
|  | **Insert** | Successfully inserts data into (all/most/some/no) of the tables: Includes own details as one of the customers; INSERT file includes primary keys. NO evidence exporting from GUI. They work properly | 5 |  |
|  | **Sufficient data** | Contains (sufficient/some/no) data to test (all/some/no) of the queries (thoroughly/ only with limited entries/no entries) | 5 |  |
| **SQL Queries** | | | **/40** |  |
|  | **Q1** | - appropriate statement-  - appropriate ORDER BY  -SQL runs well | 3 |  |
|  | **Q2** | - appropriate statement  --Includes required where clause  -correct joins  -SQL runs well | 3 |  |
|  | **Q3** | -appropriate statement formatted as per instructions (eg name and address)  -Includes required where clause  -correct joins  -SQL runs well | 4 |  |
|  | **Q4** | - appropriate statement formatted as per instructions (eg address)  -Includes required where clause  -correct joins  -SQL runs well | 3 |  |
|  | **Q5** | - appropriate statement  -Includes required where clause  -correct joins  -correct use of order by  -SQL runs well | 4 |  |
|  | **Q6** | -correct statement according to the description (AS [Total: ])  - Includes required group by  SQL runs well | 4 |  |
|  | **Q7** | -correct statement according to the description  -Includes required where clause  - Correct use of pattern matching- like-  - SQL runs well | 4 |  |
|  | **Q8** | - appropriate statement  -Includes required where clause  -correct joins  -correct use of sub-query  - SQL runs well | 5 |  |
|  | **Q9** | -correct statement according to the description  -Includes required where clause  - Includes required sub-query  - SQL runs well | 5 |  |
|  | **Q10** | -correct statement according to the description  -Includes required where clause  - SQL runs well | 3 |  |
| **ITECH5006 ONLY** | | | **15** |  |
|  | **Q1** | -correct statement  -correct calculation  -Includes required where clause  -correct joins  - SQL runs well | 4 |  |
|  | **Q2** | -correct statement  -Includes required where clause  -correct joins  - SQL runs well | 5 |  |
|  | **Q3** | -correct statement and formatting i.e. $, decimal points 2  -Includes required where clause  -correct joins  - Includes required sub-query  - SQL runs well | 6 |  |
|  | **Total ITECH 5006** | |  | **/90** |
|  | **Total ITECH1006** | |  | **/75** |
|  | **Weight** | |  | **/20%** |

**General Comment:**