C02013 Assignment 2 – Semester 1, 2016/2017

Topic A

Notes

- Students should read *everything* presented below carefully.
- This assignment is worth **20%** of the overall grade.
- Assignment 2 is group work.

Submission

You should submit your coursework to your ***lab*** tutor on Wednesday 30th November 2016.

Each group has to submit the following items:

- A paper report.
- A SQL script file.

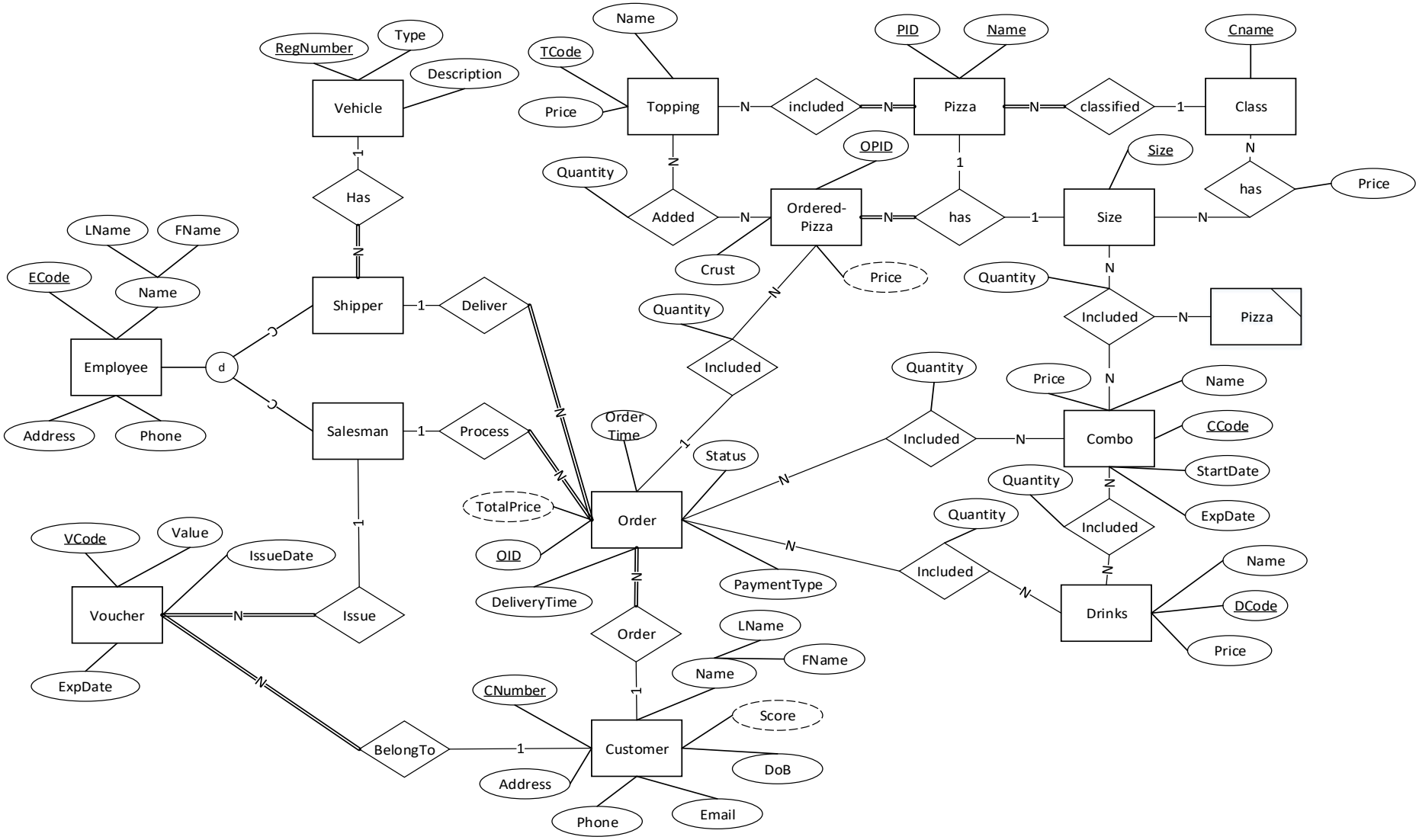
Requirements

In the assignment 2, you are expected to implement and query the database as shown in following pages, using Oracle:

Part 1: (5p)

A- Implementing the database (3.5p)

Given the following EERD and its relational database schema, students are required to implement it using ORACLE DBMS.



PIZZA			
Column name	***PK***	***FK***	***Constraints - Notes***
PID	x		Auto increment ID
Name			Unique Not Null
CName		x	CLASS.CName Not Null

CLASS			
Column name	***PK***	***FK***	***Constraints - Notes***
CName	x		

SIZE			
Column name	***PK***	***FK***	***Constraints - Notes***
Size	x		

CLASS-PRICE			
Column name	***PK***	***FK***	***Constraints - Notes***
CName	x	x	CLASS.CName
Size	x	x	SIZE.Size
Price			Not Null

TOPPING			
Column name	***PK***	***FK***	***Constraints - Notes***
TCode	x		Auto increment ID
Name			Not Null
Price			Not Null

PIZZA-TOPPING			
Column name	***PK***	***FK***	***Constraints - Notes***
PID	x	x	PIZZA.PID
TCode	x	x	TOPPING.TCode

ORDERED-PIZZA			
Column name	***PK***	***FK***	***Constraints - Notes***
OPID	x		Auto increment ID
PID		x	PIZZA.PID Not Null
Size		x	SIZE.Size Not Null
Crust			Set of acceptable values {Thin, Medium, Thick} Not Null
Price			*(*Reference: Part 1 – B – 1*)*
OID		x	ORDER.OID Not Null
Quantity			Quantity > 0 Default = 1 Not Null

ORDERED-TOPPING			
Column name	***PK***	***FK***	***Constraints - Notes***
OPID	x	x	ORDERED-PIZZA.OPID
TCode	x	x	TOPPING.TCode
Quantity			Quantity > 0 Default = 1 Not Null

DRINKS			
Column name	***PK***	***FK***	***Constraints - Notes***
DCode	x		Auto increment ID
Name			Not Null
Price			Not Null

ORDER			
Column name	***PK***	***FK***	***Constraints - Notes***
OID	x		Auto increment ID
OrderTime			Default = the CURRENT time Not Null
DeliveryTime			Delivery time must be later than the order time at least 30 minutes. Default = order time + 30 minutes Not Null
PaymentType			Set of acceptable values {Cash, Credit Card} Not Null
Status			Set of acceptable values {Done, Processing, Delivering, Cancel} Default = "Processing" Not Null
TotalPrice			*(*Reference: Part 1 – B – 2*)*
CNumber		x	CUSTOMER.CNumber Not Null
ShipperCode			EMPLOYEE.ECode ShipperCode must belong to an employee whose JobType is "Shipper" Not Null
SalesmanCode			EMPLOYEE.ECode SalesmanCode must belong to an employee whose JobType is "Salesman" Not Null

COMBO			
Column name	***PK***	***FK***	***Constraints - Notes***
CCode	x		Auto increment ID
Name			Not Null
StartDate			Not Null
ExpirationDate			(ExpirationDate >= StartDate) or (ExpirationDate = NULL) ExpirationDate = Null: The expiration date has not been determined yet. And the combo is still valid at the CURRENT time.
Price			*(*Reference: Part 1 – B – 3*)* Not Null

COMBO-PIZZA			
Column name	***PK***	***FK***	***Constraints - Notes***
CCode	x	x	ORDER.OID
PID	x	x	PIZZA.PID
Size	x	x	SIZE.Size
Quantity			Quantity > 0 Default = 1 Not Null

COMBO-DRINKS			
Column name	***PK***	***FK***	***Constraints - Notes***
CCode	x	x	ORDER.OID
DCode	x	x	DRINKS.DCode
Quantity			Quantity > 0 Default = 1 Not Null

ORDER-DRINKS			
Column name	***PK***	***FK***	***Constraints - Notes***
OID	x	x	ORDER.OID
DCode	x	x	DRINKS.DCode
Quantity			Quantity > 0 Default = 1 Not Null

ORDER-COMBO			
Column name	***PK***	***FK***	***Constraints - Notes***
OID	x	x	ORDER.OID
CCode	x	x	COMBO.CCode
Quantity			Quantity > 0 Default = 1 Not Null

CUSTOMER			
Column name	***PK***	***FK***	***Constraints - Notes***
CNumber	x		Auto increment ID
FName			Not Null
LName			Not Null
DoB			
Email			A valid email format. *(Simple email format: a string of any number of [0-9], [a-z], [A-Z] , '.' , '_' and a single '@')*
Phone			It is a string of 10 or 11 digits and starts with the digit '0' Not Null
Address			Not Null
Score			Default = 0 Not Null *(*Reference: Part 1 – B – 4*)*

EMPLOYEE			
Column name	***PK***	***FK***	***Constraints - Notes***
ECode	x		If an employee is a salesman, his/her ECode must obey the pattern "SM[0-9]^3". The number following "SM" is automatically increased. Example: SM001, SM002,… If an employee is a shipper, his/her ECode must obey the pattern: "SP[0-9]^3". The number following "SP" is automatically increased. Example: SP001, SP002,…
FName			Not Null
LName			Not Null
Phone			It is a string of 10 or 11 digits and starts with the digit '0' Not Null
Address			Not Null
IsSalesMan			0: No 1: Yes Not Null
IsShipper			0: No 1: Yes Not Null
RegNumber		x	VEHICLE.RegNumber If IsShipper = 1, RegNumber is Not Null. If IsShipper = 0, RegNumber must be Null.

VEHICLE			
Column name	***PK***	***FK***	***Constraints - Notes***
RegNumber	x		
Type			
Description			

VOUCHER			
Column name	***PK***	***FK***	***Constraints - Notes***
VCode	x		Auto increment ID
Value			Not Null
IssueDate			Not Null
ExpirationDate			ExpirationDate > IssueDate Not Null
ECode		x	EMPLOYEE.ECode Not Null
CNumber		x	CUSTOMER.CNumber Not Null

B- Constraints (1p)

Write triggers to ensure the following constraints or business rules:

1. In table ORDERED-PIZZA, Price is the sum of prices of an ordered pizza and all added toppings.
2. In table ORDER, TotalPrice is the sum of prices of pizzas, drinks, combos included in an order.
3. The combo price must be less than the total price of all individuals in the combo.
4. For a successful order (ORDER.Status = "Done"), the score is accumulated some points which is calculated as Floor(Order's price / 100.000đ).
   Example: Floor(550.000 / 100.000) = 5

(*) If you have any questions, please consult your lab tutor.

C- Index (0.5p)

Database users often query the following information:

1. Get pizza's information by its name.
2. Get customer's information by his/her name.

Based on the common queries, you should create the index on the proper fields in order to increase the system performance.

Part 2: INSERT, UPDATE, DELETE, SELECT (2p)

1. You are required to insert valid and meaningful data into the database. Each table has at least 4 rows. (1p)
2. Change the price of the drink whose name is "Coca-cola" to 22.000. (0.25p)
3. Delete a salesman whose name is "Nguyễn Văn Ân". (0.25p)
4. Retrieve all pizzas in class "Premium". (0.25p)
5. Calculate the total value (TotalPrice) of all orders which include a combo "Happy Friday" in August 2016. (0.25p)

Part 3: STORE PROCEDURE, FUNCTION (2p)

1. Create a procedure/function to insert into table EMPLOYEE.
   Input: FName, LName, Phone, Address, JobType, RegNumber (if he/she is a shipper)
   Output: SUCCESSFUL or FAIL. If FAIL, show the error message. (1p)
2. Create a procedure/function to find top three best-seller pizzas in a period of time.
   Input: Start time, End time. If End time is Null, set it to CURRENT time
   Output: List PIDs and Names of top three best-seller pizzas. (1p)

Part 4: ACCESS CONTROL (1p)

1. Login to the database with a DBA user (SYS, SYSTEM, SYSMAN).
2. Create 2 users:
   - User *MenuUser*.
   - User *SalesUser*.
3. Grant the 2 users the privilege/role to connect to the database.
4. *MenuUser* is responsible for all items in the menu. Grant proper privileges to *MenuUser*.
5. *SalesUser* is responsible for orders, customers, vouchers, assigning shipper for an order. However, he/she cannot modify information of items in the menu, employees. Grant proper privileges to *MenuUser*.
6. Disallow *SalesUser* to DELETE on CUSTOMER.