CSE2DBF – CSE4DBF

Final Exam Information and Sample Exam solution

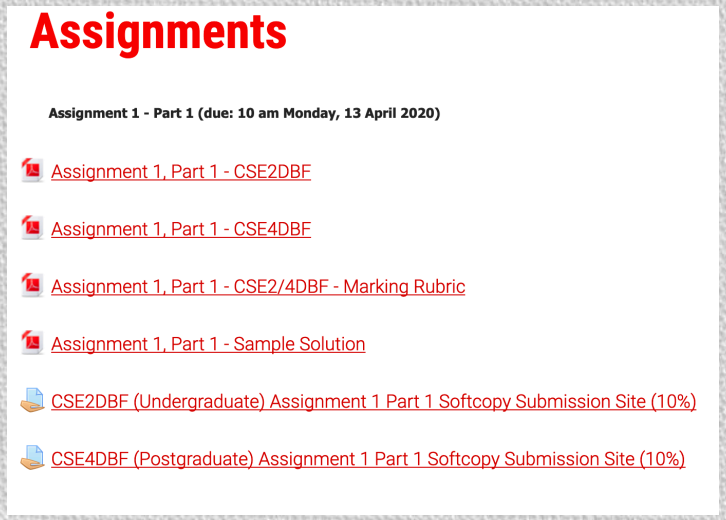
01/06/2020

Exam Details

- **Venue**:
  - LMS online

  I will create an EXAM site in LMS which is similar to the Assignment site. You can download the exam paper and submit your exam to this site. (A detailed specification will be provided)

- Date: 23/06/2020
- Time starts: 9:00am
- Time ends: 2:15pm
- Exam Duration: 5 hours 15 mins



Exam Structure

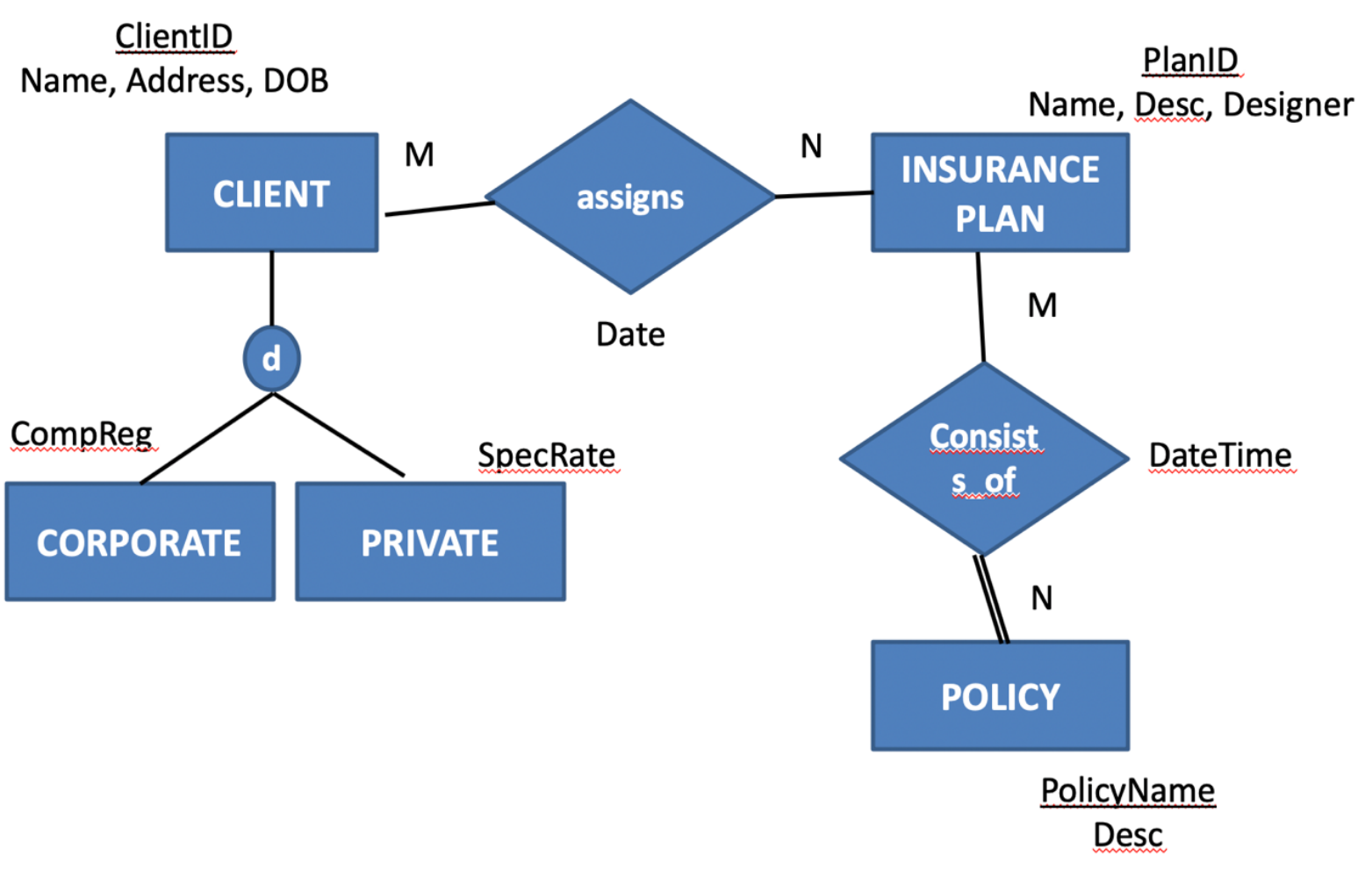
- Nine questions, structured following the topics in this subject.
  - Five single-choice questions and four descriptive/analytical questions (Total 180 marks)
  - Q1: ER/EER diagram (single-choice question, 10 marks)
  - Q2: ER/EER diagram (single-choice question, 10 marks)
  - Q3: ER/EER diagram (single-choice question, 10 marks)
  - Q4: Relational Algebra (single-choice question, 10 marks)
  - Q5: Relational Algebra (single-choice question, 10 marks)
  - Q6: Transformation of EER to relational tables (25 marks=20+5)
  - Q7: Normalization (30 marks)
  - Q8: SQL (30 marks = 6 * 5 subquestions)
  - Q9: PL/SQL (Stored Procedure/Function/Trigger) (45 marks = 15+15+15)
- You need to answer ALL questions. There is no optional question.

Exam Consultation

- Zoom: https://latrobe.zoom.us/j/973092551
  - 8 Jun 2020: 1-2pm
  - 12 Jun 2020: 1-2pm
  - 15 Jun 2020: 1-2pm
  - 19 Jun 2020: 1-2pm

  I will set up the “EXAM Consultation Selection” in LMS.

Q1


ClientID
Name, Address, DOB
PlanID
Name, Desc, Designer
CLIENT
M
assigns
N
INSURANCE PLAN
Date
M
d
CompReg
SpecRate
Consist s of
DateTime
CORPORATE
PRIVATE
N
POLICY
PolicyName
Desc

Q4

$$TempList1 \leftarrow \pi_{BookIDd,\ Title,\ Price,\ Quantity}(Book\_T) \overset{\cap}{\bowtie}_{BookID\ =\ BookID} (Order\_T)$$

$$TempList2 \leftarrow \pi_{AuthorID,\ Quantity}(Book\_Author\_T) \bowtie_{BookID\ =\ BookID} (TempList1)$$

$$RESULT \leftarrow \pi_{FName,\ LName}(Author\_T) \bowtie_{AuthorID\ =\ AuthorID} (TempList2)$$

TempList1

BookID	Title	Price	Quantity
B1	The Godfather	30	10
B1	The Godfather	30	15
B2	The Alchemist	15	NULL
B3	The Last Don	35	20
B4	Falling Angels	25	20
B4	Falling Angels	25	30
B5	Portrait in Sepia	35	10

TempList2

AuthorID	Quantity
5	10
5	15
1	NULL
5	20
4	20
4	30
2	10

RESULT

FName	LName
Paul	Coelho
Isabel	Allende
Tracy	Chevalier
Tracy	Chevalier
Mario	Puzo
Mario	Puzo
Mario	Puzo

Q5

$Temp1 \leftarrow \pi_{roomNo}(\sigma_{name = \text{'Hilton Melbourne'}}(Hotel) \bowtie_{hotelNo = hotelNo} (Room))$
$Temp2 \leftarrow \pi_{roomNo}((\sigma_{name = \text{'Hilton Melbourne'}}(Hotel)) \bowtie_{hotelNo = hotelNo} (\sigma_{DateFrom <= SYSDATE \text{ AND } DateTo >= SYSDATE}(Booking))$
$Results \leftarrow Temp1 \cap Temp2$

Will retrieve the rooms at "Hilton Melbourne" that are UNAVAILABLE.

$Temp1 \leftarrow \pi_{roomNo}(\sigma_{name = \text{'Hilton Melbourne'}}(Hotel) \bowtie_{hotelNo = hotelNo} (Room))$
$Temp2 \leftarrow \pi_{roomNo}(\sigma_{DateFrom <= SYSDATE \text{ AND } DateTo >= SYSDATE}(Booking))$
$Results \leftarrow Temp1 \cap Temp2$

Temp2 will return all UNAVAILABLE room in any hotel in the database.

$Temp1 \leftarrow \pi_{roomNo}(\sigma_{name = \text{'Hilton Melbourne'}}(Hotel) \bowtie_{hotelNo = hotelNo} (Room))$
$Temp2 \leftarrow \pi_{roomNo}((\sigma_{name = \text{'Hilton Melbourne'}}(Hotel)) \bowtie_{hotelNo = hotelNo} (\sigma_{DateFrom <= SYSDATE \text{ AND } DateTo >= SYSDATE}(Booking))$
$Results \leftarrow Temp1 - Temp2$

Will retrieve the rooms at "Hilton Melbourne" that are AVAILABLE.

$Temp1 \leftarrow \pi_{roomNo}(\sigma_{name = \text{'Hilton Melbourne'}}(Hotel) \bowtie_{hotelNo = hotelNo} (Room))$
$Temp2 \leftarrow \pi_{roomNo}(\sigma_{DateFrom <= SYSDATE \text{ AND } DateTo >= SYSDATE}(Booking))$
$Results \leftarrow Temp1 - Temp2$

Temp2 will return all UNAVAILABLE rooms in any hotel in the database. So, the DIFFERENCE operation might create wrong result.

Question 6

STEP 1:
BOOK (BookID, Title, NumberPages, ReleaseDate)
WRITER (WriterID, WriterName, WriterAddress, CountryofOrigin)
AGENT (AgentID, AgentName, AgentAddress, AgentContact)
EDITOR (EditorID, EditorName, EditorContact, ForeignLang?)
PLATFORM (PlatformName, PlatformDesc)

STEP 2:
BOOKCHAPTER (*BookID*, ChapterNo, Excerpt)

STEP 3: No 1-1 Relationship

STEP 4:
BOOK (BookID, Title, NumberPages, ReleaseDate, *EditorID*)
WRITER (WriterID, WriterName, WriterAddress, CountryofOrigin, *AgentID*)
EDITOR (EditorID, EditorName, EditorContact, ForeignLang?, *MentorEditor*)

STEP 5:
WRITES (*WriterID*, *BookID*)

STEP 6:
BOOKGENRE (*BookID*, Genre)

STEP 7: No ternary relationship.

Question 6

STEP 8:

8A:
PRINTED (*BookID*, NumberofCopies, PrintedPrice)
E-BOOK (*BookID*, EBookPrice)
Or,
8D
BOOK (BookID, Title, NumberPages, ReleaseDate, EditorID, PrintedFlag, NumberofCopies, PrintePrice, EBookFlag, EBookPrice)

REPEAT STEP 2-7

STEP 5:

VIEWEDON (*BookID*, *PlatformName*)

FINAL TABLES (Using 8A):

AGENT (AgentID, AgentName, AgentAddress, AgentContact)
PLATFORM (PlatformName, PlatformDesc)
BOOKCHAPTER (*BookID*, ChapterNo, Excerpt)
BOOK (BookID, Title, NumberPages, ReleaseDate, *EditorID*)
WRITER (WriterID, WriterName, WriterAddress, CountryofOrigin, *AgentID*)
EDITOR (EditorID, EditorName, EditorContact, ForeignLang?, *MentorEditor*)
WRITES (*WriterID*, *BookID*)
BOOKGENRE (*BookID*, Genre)
PRINTED (*BookID*, NumberofCopies, PrintedPrice)
E-BOOK (*BookID*, EBookPrice)
VIEWEDON (*BookID*, *PlatformName*)

Question 7a

UNF:
Bill(ProdNo, ProdDesc, (PartNo, PartDesc, QuantityUsed, Location, Code))

1NF:
Product(**ProdNo**, ProdDesc)
Bill(***ProdNo***, **PartNo**, PartDesc, QuantityUsed, Location, Code)

2NF:
Part(**PartNo**, PartDesc, Location, Code)
Bill(***ProdNo***, ***PartNo***, QuantityUsed)

3NF, BCNF: Already in 3NF and BCNF

FINAL
Product(**ProdNo**, ProdDesc)
Part(**PartNo**, PartDesc, Location, Code)
Bill(***ProdNo***, ***PartNo***, QuantityUsed)

Question 7b

UNF:

Customer (customerNo, customerName, gender, address, city, state, postcode, dob, contactNumber, email, (featurecategory, (feature)))

1NF:

Customer (customerNo, customerName, gender, address, city, state, postcode, dob, contactNumber, email)

FeatureCategory (featureCategoryID, featureCategory)

DesiredFeatures(featureID, feature, *featureCategory*, *customerNo*)

2NF:

DesiredFeatures (*customerNo*, *featureID*)

FeatureCategory (featureCategoryID, featureCategory)

Features (featureID, feature, *featureCategoryID*)

3NF, BCNF: Already in 3NF and BCNF

FINAL

Customer (customerNo, customerName, gender, address, city, state, postcode, dob, contactNumber, email)

DesiredFeatures (*customerNo*, *featureID*)

FeatureCategory (featureCategoryID, featureCategory)

Features (featureID, feature, *featureCategoryID*)

Question 8

a)

```
SELECT Country, COUNT (*)
FROM MEMBER
GROUP BY Country
ORDER BY count(*) DESC;
```

b)

```
SELECT c.categoryId, c.categoryName, c.categoryDescription
FROM category c, item i
WHERE c.categoryId = i.categoryId
GROUP BY c.categoryId, c.categoryName, c.categoryDescription
HAVING COUNT(c.categoryId) = (SELECT MAX(COUNT(categoryId))
                              FROM item
                              GROUP BY categoryId);
```

c)

```
SELECT m.username, m.firstName, m.lastName, m.address,
m.city, m.country, m.postcode, m.phoneAH, m.phoneBH, m.email
FROM MEMBER M
WHERE m.username NOT IN
                (SELECT i.sellerUsername
                 FROM ITEM i, AUCTION a
                 WHERE  i.itemNumber = a.itemNumber);
```

Question 9

d)

```
SELECT I.itemNumber, I.itemName, B.bidAmount
FROM AUCTION A, BID B, CATEGORY C, ITEM I
WHERE I.itemNumber = A.itemNumber
AND A.auctionNumber = B.auctionNumber
AND I.categoryid = C.categoryid
AND B.bidAmount > A.reserveamount
AND A.itemSold = 'N'
AND C.categoryname = 'Pottery'
ORDER BY I.itemNumber;
```

Question 5

a)

```
CREATE OR REPLACE PROCEDURE CustomerOrders (CustomerNum NUMBER)
AS
    CURSOR orderCursor IS
    SELECT O.Order_no, O.Order_date,
    SUM (P.Prod_price * OD.Order_qty) AS totalPrice
    FROM   Orders O, Order_Details OD, Products P
    WHERE  CustomerNum = O.CustNo
    AND O.Order_no = OD.Order_no
    AND OD.Prod_cod = P.Prod_cod
    GROUP BY O.Order_no, O.Order_date;
BEGIN
    FOR orderPointer IN orderCursor LOOP

    DBMS_OUTPUT.PUT_LINE (orderPointer.Order_no || ' '
    || orderPointer.Order_date || ' ' ||
    orderPointer.totalPrice);

    END LOOP;
END CustomerOrders;
/
```

Question 5

b)

```
CREATE OR REPLACE TRTIGGER New_Order
BEFORE INSERT ON ORDERS_DETAILS
FOR EACH ROW

BEGIN

UPDATE PRODUCTS
SET Qty_on_hand = Qty_on_hand - :new.order_qty
WHERE Prod_cod = :new.Prod_cod;

END New_Order;
/
```