2024

WRDV301 Portfolio

Hokwana, Oyintanda Samkele, 217612792

5/27/2024

Contents

Rubric

Class activities

CLASS ACTIVITY	YOUR MARK	MAX MARK	CLASS AVG	COMMENT
01 – 15 FEB*	y	n/a	n/a	Attendance – nothing to include
02 – 29 FEB	3	6	4/6	Moodle Quiz – not included
03 – 14 MAR	10	10	8/10	Include in portfolio
04 – 14 MAR*	y	n/a	n/a	Attendance – nothing to include
05 – 11 APR	5	5	5/5	Include in portfolio
06 – 18 APR	9	15	6/15	Include in portfolio
07 – 2 MAY	0	9	6/9	Include in portfolio

* Indicating attendance; no mark allocated

Assignments

ASSIGNMENT	YOUR MARK	MAX MARK	CLASS AVG	RESUBMIT (MAX OF ONE) a*	RESUBMIT (MAX OF ONE) Mark	FIRST SUBMIT (MAX OF ONE) a*	FIRST SUBMIT (MAX OF ONE) Mark
01 – 4 MAR	62	66	54/66	N		N	
02 – 18 MAR	28	36	28/36	Y		N	
03 – 15 APR	27	27	25/27	N		N	
04 – 29 APR	41	45	38/45	N		N	
05 – 13 MAY	27	27	19/27	N		N	

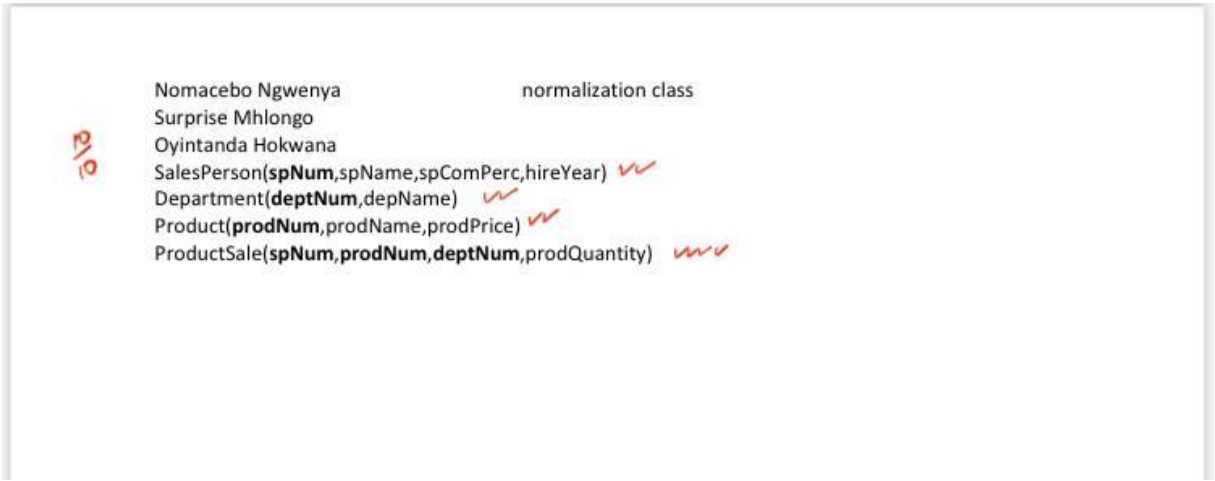
* Indicate with awhere relevant.

Portfolio

Overview Component	Does the criterion meet the expectations? 0 No	1 Partially	2 Yes	3 Exceeds	Score
1. Submitted artefacts	Required sections or artefacts not included	Some sections and some artefacts included	All sections and all artefacts included	All sections and all artefacts appropriately included and documented	
2. Evidence of learning	Little or no evidence of learning	Some evidence of learning, but not sufficient	Adequate evidence of learning	Clear evidence of learning	
3. Evidence of reflection	Little or no evidence of reflection	Some evidence of reflection, but not sufficient	Adequate evidence of reflection	Clear evidence of reflection	
4. Presentation	Instructions not followed; many errors	Most instructions followed; still some errors	Well organised; instructions followed	Well organised, well structured, all instructions followed	

Class Submissions

03-14 MAR (Class Activity)



Nomacebo Ngwenya normalization class
Surprise Mhlongo
Oyintanda Hokwana
SalesPerson(**spNum**,spName,spComPerc,hireYear)
Department(**deptNum**,depName)
Product(**prodNum**,prodName,prodPrice)
ProductSale(**spNum**,**prodNum**,**deptNum**,prodQuantity)

05-11 APR (Class Activity)

Rubushe Bathandwa
Oyintanda Hokwana
Kavier Jogie

WRDV301 Class activity

11 April 2024

5/5

Did you find the killer?

```
1 INSERT INTO solution VALUES (1, 'Jeremy Bowers');
2
3       SELECT value FROM solution;
```

RUN ⇩ RESET

value
Congrats, you found the murderer! But wait, there's more... If you think you're up for a challenge, try querying the interview transcript of the murderer to find the real villain behind this crime. If you feel especially confident in your SQL skills, try to complete this final step with no more than 2 queries. Use this same INSERT statement with your new suspect to check your answer.

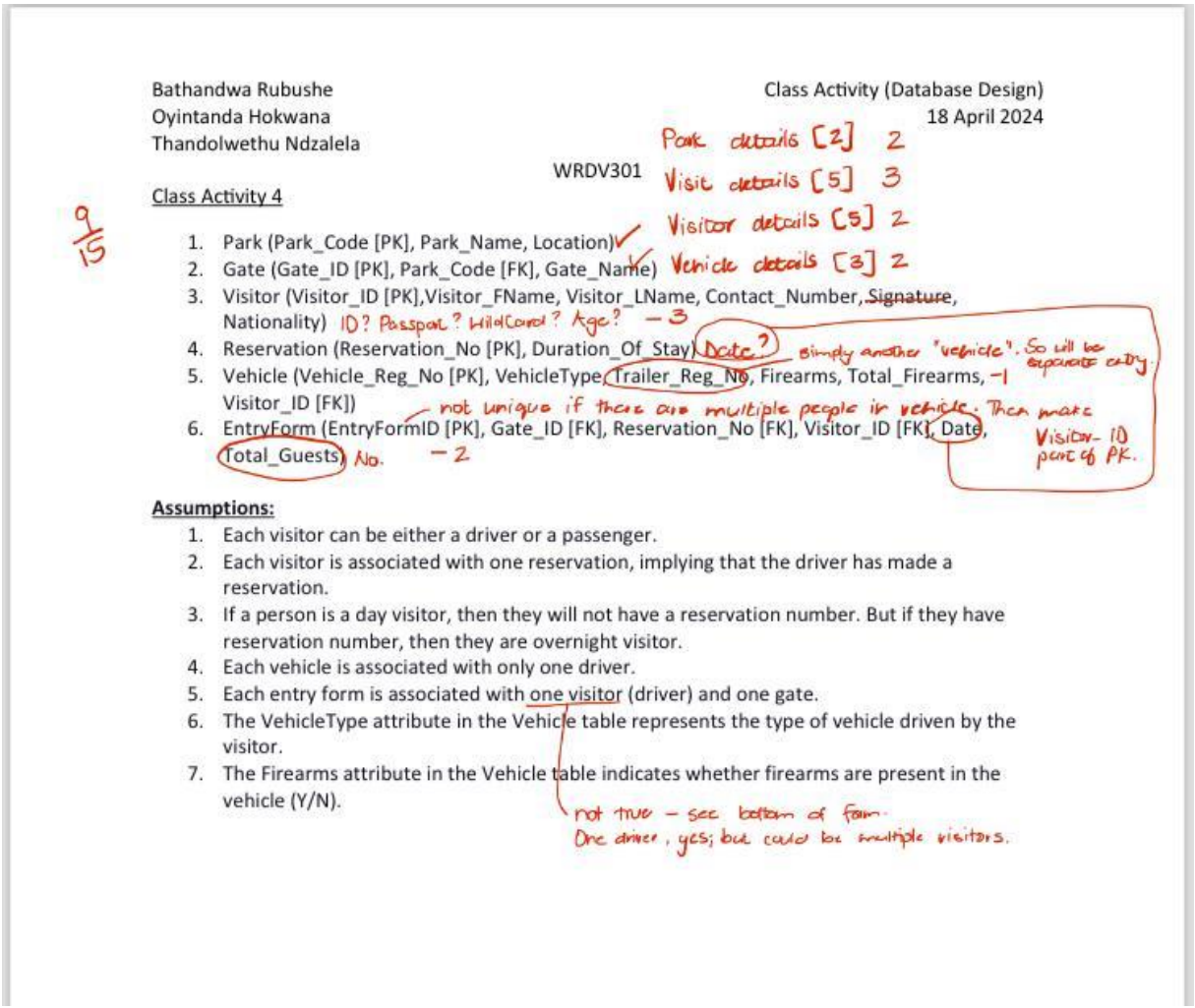
Well done.

```
1 INSERT INTO solution VALUES (1, 'Miranda Priestly');
2
3       SELECT value FROM solution;
```

RUN ⇩ RESET

value
Congrats, you found the brains behind the murder! Everyone in SQL City hails you as the greatest SQL detective of all time. Time to break out the champagne!

06-18 APR (Class Activity)



Bathandwa Rubushe
Oyintanda Hokwana
Thandolwethu Ndzalela

Class Activity (Database Design)
18 April 2024

WRDV301

Park details [2] 2
Visit details [5] 3
Visitor details [5] 2
Vehicle details [3] 2

9/15

Class Activity 4

1. Park (Park_Code [PK], Park_Name, Location)
2. Gate (Gate_ID [PK], Park_Code [FK], Gate_Name)
3. Visitor (Visitor_ID [PK],Visitor_FName, Visitor_LName, Contact_Number, ~~Signature~~, Nationality) ID? Passport? HildCard? Age? −3
4. Reservation (Reservation_No [PK], Duration_Of_Stay) Date?
5. Vehicle (Vehicle_Reg_No [PK], VehicleType, Trailer_Reg_No, Firearms, Total_Firearms, Visitor_ID [FK]) simply another "vehicle". So will be separate entry. −1
6. EntryForm (EntryFormID [PK], Gate_ID [FK], Reservation_No [FK], Visitor_ID [FK], Date, Total_Guests) not unique if there are multiple people in vehicle. Then make Visitor_ID part of PK. No. −2

Assumptions:

1. Each visitor can be either a driver or a passenger.
2. Each visitor is associated with one reservation, implying that the driver has made a reservation.
3. If a person is a day visitor, then they will not have a reservation number. But if they have reservation number, then they are overnight visitor.
4. Each vehicle is associated with only one driver.
5. Each entry form is associated with one visitor (driver) and one gate.
6. The VehicleType attribute in the Vehicle table represents the type of vehicle driven by the visitor.
7. The Firearms attribute in the Vehicle table indicates whether firearms are present in the vehicle (Y/N).

not true – see bottom of form.
One driver, yes; but could be multiple visitors.

07-02 MAY(Class Activity)

There was no submission for this activity.

Assignments

Assignment 1

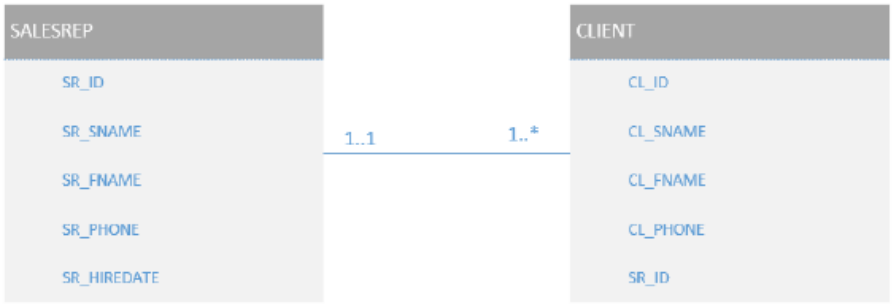
Draft

Hokwana Oyintanda
Ngwenya Nomacebo ***Assignment1 Draft*** ***26/02/2024.***

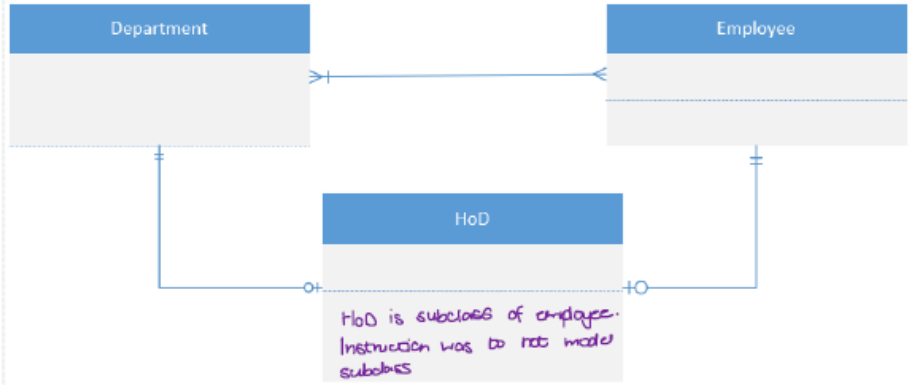
1. Ww
   a) A SALESREP must serve at least 1 CLIENT.
   A CLIENT can only be served by one SALESREP.
   One CLIENT can be served by how many SALESREP?
   - One and only one

   One SALESREP can serve how many CLIENTS?
   - One or more

   b)



2. W



3. a)

Table	Primary Key(s)	Foreign key(s)
CHARTER	CH_TRIP	CUS_CODE, AC_NUMBER
AIRCRAFT	AC_NUMBER	MOD_CODE
MODEL	MOD_CODE	N/A
EMPLOYEE	EMP_NUM	N/A
PILOT	N/A	EMP_NUM
CUSTOMER	CUS_CODE	N/A

Missing FKs

Missing PK

footer ?

Hokwana Oyintanda
Ngwenya Nomacebo ***Assignment1 Draft*** ***26/02/2024.***

How does it then exhibit referential integrity if there is no FK?

b) Yes. There is no Foreign Key in MODEL entity meaning the MODEL table does not depend on other tables/ not referenced on other tables.

c) No. The PILOT entity does not have a PRIMARY KEY. Not true. What is a PK's characteristics?

4. Efdjijf

a)

stuNum
200075304
219075320
219707527
220075282
221075312
221075339
227075266

✓

b)

stuNum	lectDept
200075304	ACC
219075920	BM
219707527	STATS
220075282	STATS
221075312	CS

✓

5.

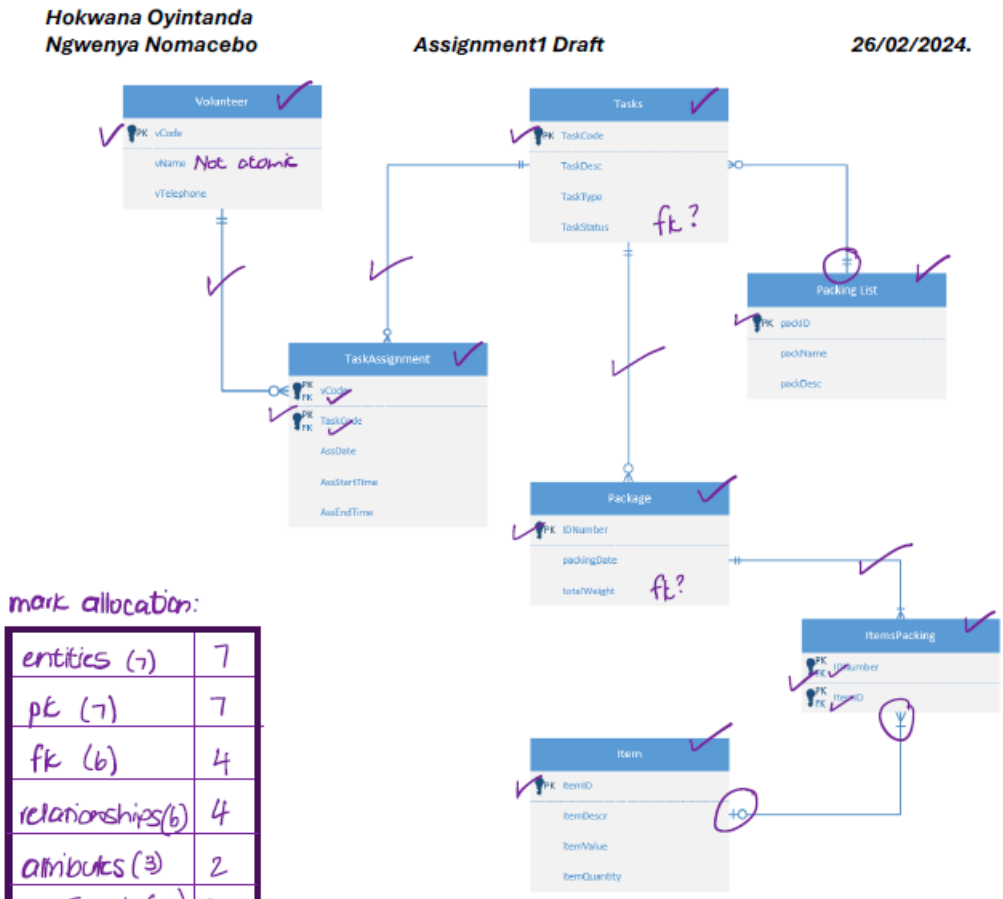
a)

i) Unary Relationship for CONTRACTOR entity – This means that the CONTRACTOR can hire themselves. They can give themselves a contract. Not quite.

ii) Binary-Relationship between CONTRACTOR entity and LABOURER entity – This is shown by the association between the CONTRACTOR entity and LABOURER entity. This is shown by a "CONTRACTOR who hires one or more LABOURERs, and each LABOURER being hired by one and only one CONTRACTOR" relationship. ✓

b) Weak entity – The LABOURER entity is weak entity because its PK is partially derived from the CONTRACTOR, this means that it existent-dependent on CONTRACTOR therefore can't exist on its own. ✓

6.



mark allocation:

entities (7)	7
pk (7)	7
fk (6)	4
relationships(6)	4
attributes (3)	2
Total (29)	24

Final

62/66

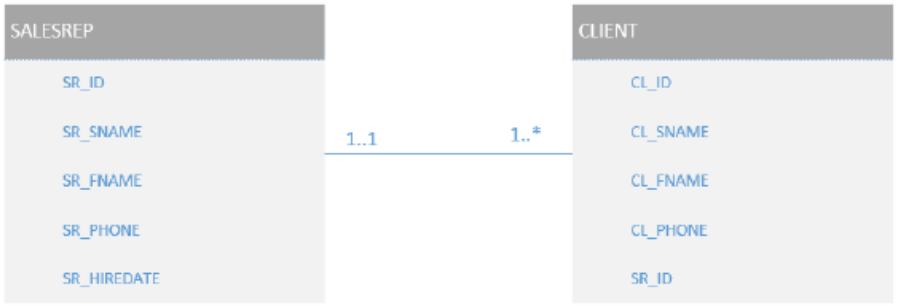
Hokwana Oyintanda
Ngwenya Nomacebo ***Assignment1Final*** ***04/03/2024.***

1. Ww
   a) A SALESREP must serve at least 1 CLIENT.
   A CLIENT can only be served by one SALESREP.
   One CLIENT can be served by how many SALESREP?
   - One and only one

   One SALESREP can serve how many CLIENTS?
   - One or more

   [7] ✓ 3

   b)



✓ 4

2.

[4]

4

HoD is an employee. Simply add the relationship you have here to the other diagram (thus you have 2 relationships)

3. a)

[10]

Table	Primary Key(s)	Foreign key(s)
CHARTER	CH_TRIP	CUS_CODE, AC_NUMBER, CH_COPILOT, CH_PILOT
AIRCRAFT	AC_NUMBER	MOD_CODE
MODEL	MOD_CODE	N/A
EMPLOYEE	EMP_NUM	N/A
PILOT	EMP_NUM	EMP_NUM
CUSTOMER	CUS_CODE	N/A

✓ ✓ 8

   b) Yes. The relationship between the MODEL and AIRCRAFT tables is established through the MOD_CODE foreign key in the AIRCRAFT table. Each MOD_CODE entry in the AIRCRAFT table corresponds to a valid entry in the MODEL table, ensuring referential integrity within the MODEL table.

Where is the fk located?

0

   c) Yes. Entity integrity is maintained across all tables by ensuring that primary keys are unique and non-null. ✓ 2

4. a)

1

Hokwana Oyintanda
Ngwenya Nomacebo ***Assignment1Final*** ***04/03/2024.***

[6]

stuNum
200075304
219075320
219707527
220075282
221075312
221075339
227075266

✓ 2

b)

stuNum	lectDept
200075304	ACC
219075920	BM
219707527	STATS
220075282	STATS
221075312	CS

✓ 4

5.

a)

[6]

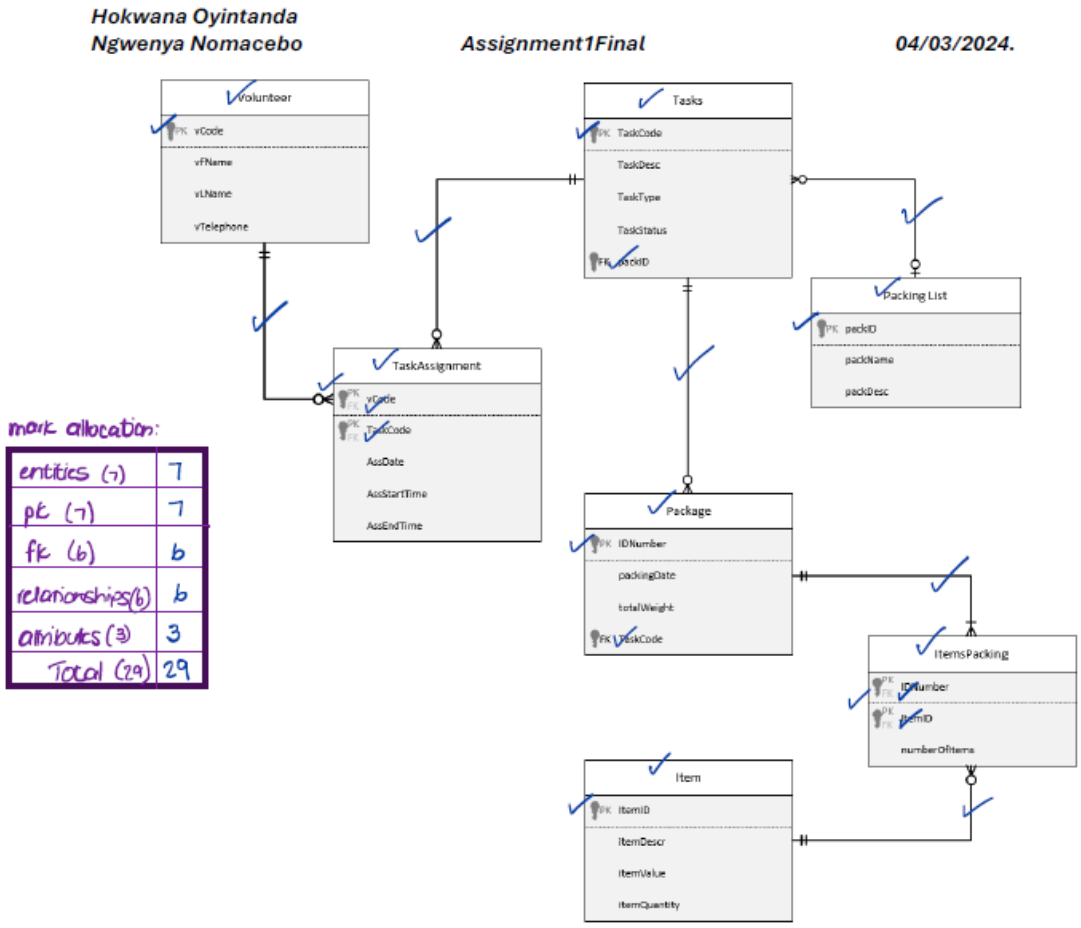
i) Unary Relationship for CONTRACTOR entity – Contractors may engage in sub-contracting relationships with other contractors for various reasons such as workload distribution, specialization, or project requirements. ✓

ii) Binary-Relationship between CONTRACTOR entity and LABOURER entity – This is shown by the association between the CONTRACTOR entity and LABOURER entity. This is shown by a "CONTRACTOR who hires one or more LABOURERs, and each LABOURER being hired by one and only one CONTRACTOR" relationship. ✓ 4

b) Weak entity – The LABOURER entity is weak entity because its PK is partially derived from the CONTRACTOR, this means that it existent-dependent on CONTRACTOR therefore can't exist on its own. ✓ 2

6.

[29]


Hokwana Oyintanda
Ngwenya Nomacebo
Assignment1Final
04/03/2024.
Voluntoer
vCode
vFName
vLName
vTelephone
Tasks
TaskCode
TaskDesc
TaskType
TaskStatus
packID
Packing List
packID
packName
packDesc
TaskAssignment
vCode
TaskCode
AssDate
AssStartTime
AssEndTime
Package
IDNumber
packingDate
totalWeight
TaskCode
ItemsPacking
IDNumber
ItemID
numberOfItems
Item
ItemID
ItemDescr
ItemValue
ItemQuantity
mark allocation:
entities (7) 7
pk (7) 7
fk (6) 6
relationships(6) 6
attributes (3) 3
Total (29) 29

Assignment 2

Draft

HokwanaOS
NgwenyaN Assignment2 Draft 2024/03/11

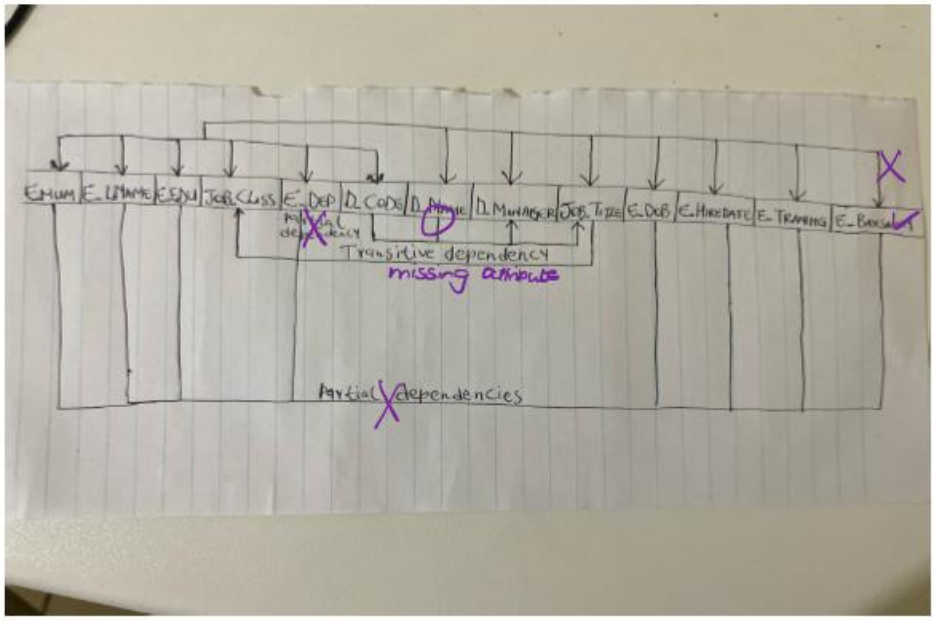
1.

2.

a) Table (**E_NUM**,E_LNAME,E_EDU,JOB_CLASS,E_DEP, **D_CODE** , D_NAME , D_MANAGER ,JOB_TITLE,E_DOB,E_HIREDATE,E_TRAINING,E_BASESALARY)

HokwanaOS
NgwenyaN Assignment2 Draft 2024/03/11

b)



3. A_TABLE (**A**, D, F)
   B_TABLE (**B**, C)
   E_TABLE (**E**, G)

Not dependency diagram.

Final

28/36

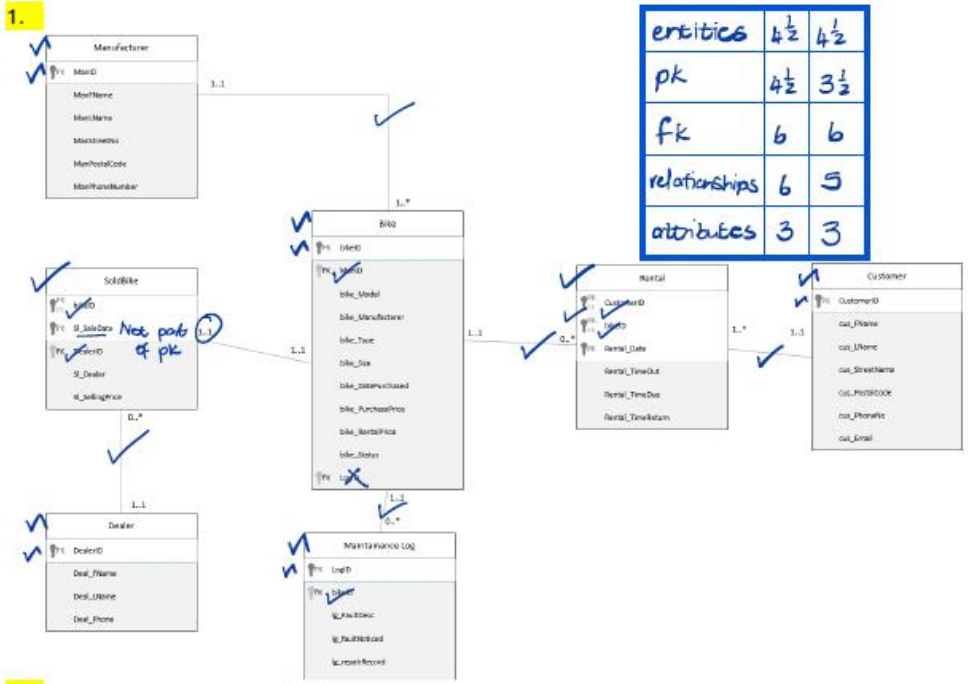
HokwanaOS
NgwenyaN

Assignment2 Final

2024/03/18

[22]

1.



entities	4½	4½
pk	4½	3½
fk	6	6
relationships	6	5
attributes	3	3

2.

a) Table (**E_NUM**, **D_CODE**, E_LNAME, E_EDU,JOB_CLASS,E_DEP, D_NAME , D_MANAGER ,JOB_TITLE,E_DOB,E_HIREDATE,E_TRAINING,E_BASESALARY)

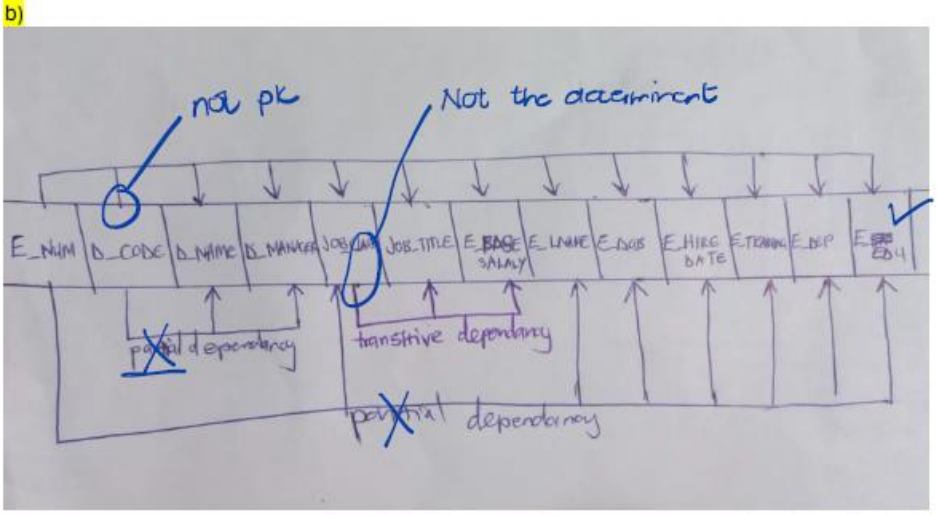
not pk

[3]

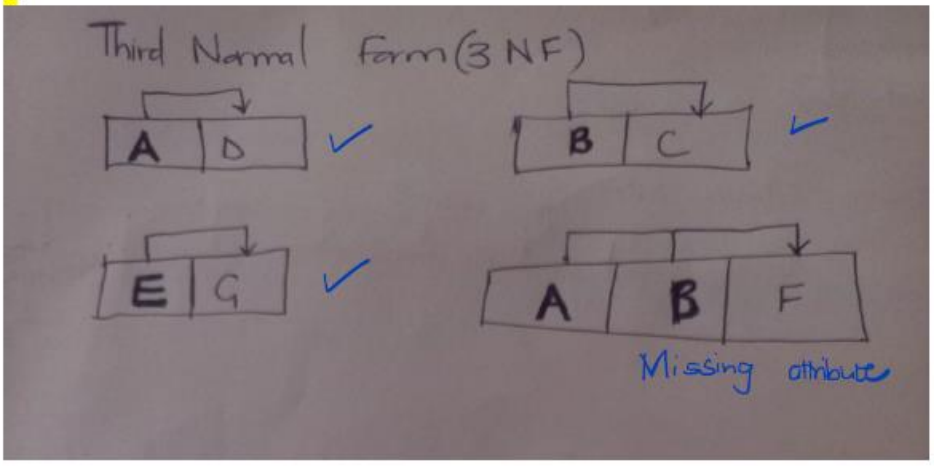
2

HokwanaOS
NgwenyaN Assignment2 Final 2024/03/18

b)

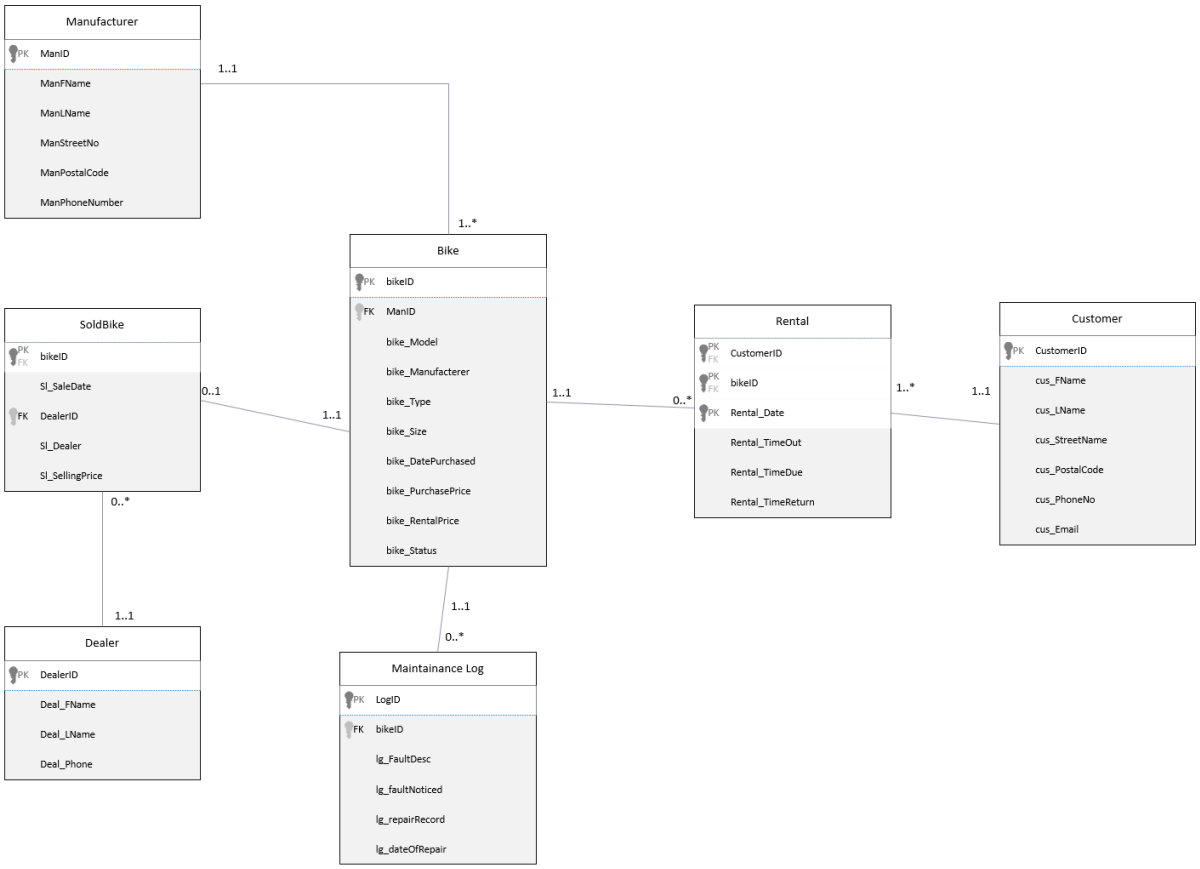


3. A



Resubmission

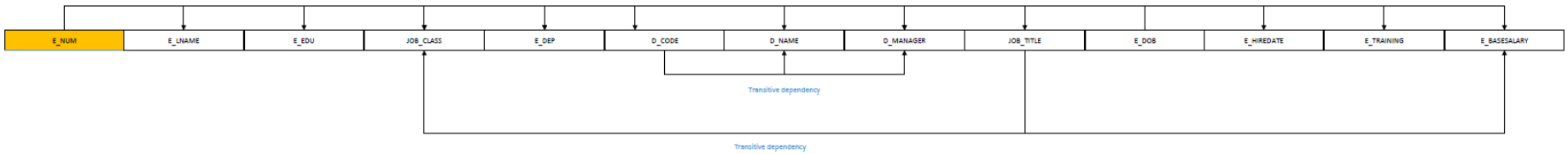
1.



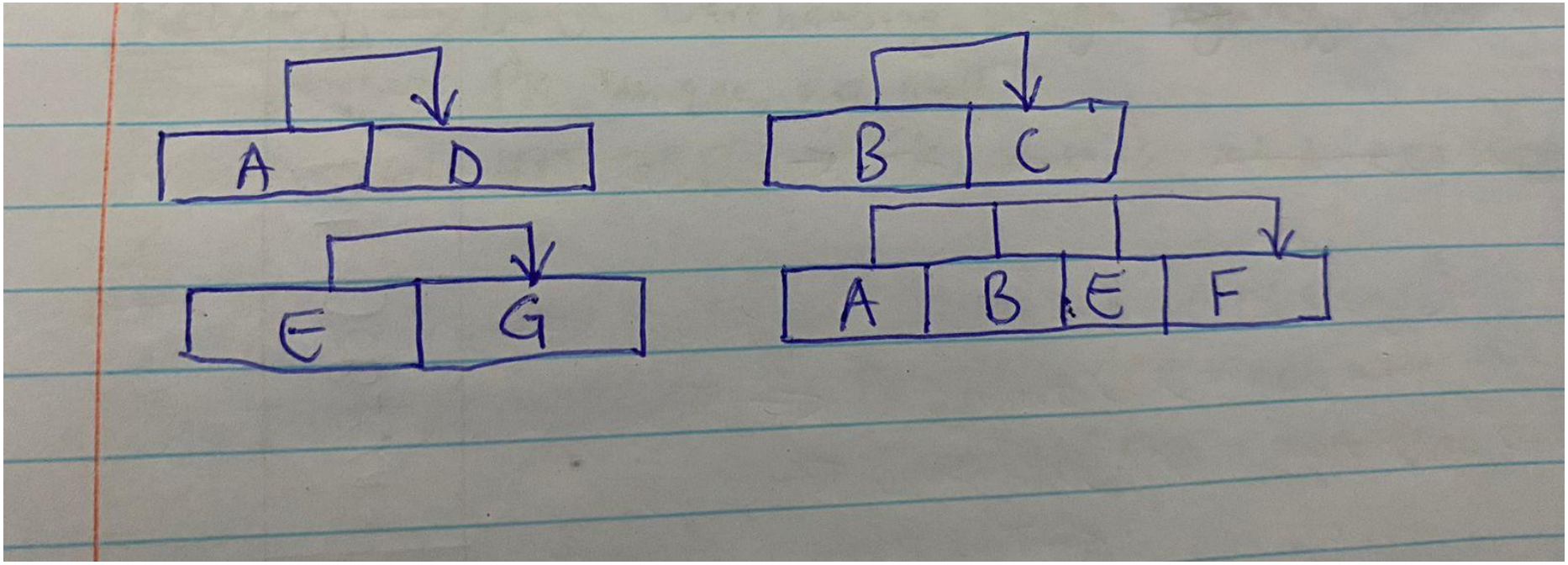
2.

a) Table (**E_NUM**, D_CODE, E_LNAME, E_EDU,JOB_CLASS,E_DEP, D_NAME , D_MANAGER ,JOB_TITLE,E_DOB,E_HIREDATE,E_TRAINING,E_BASESALARY)

b)


E_NUM
E_LNAME
E_EDU
JOB_CLASS
E_DEP
D_CODE
D_NAME
D_MANAGER
JOB_TITLE
E_DOB
E_HIREDATE
E_TRAINING
E_BASESALARY
Transitive dependency
Transitive dependency

3.


A
D
B
C
E
G
A
B
E
F

Reflection:
I've realized that I made an error in my initial understanding of implementing inheritance in this context. Initially, I assumed that the relationship was always strictly one-to-one. However, in this specific case, the relationship between Bike and SoldBike is actually zero-to-one (0..1). This is because a bike can either be scrapped or sold, but not both. Therefore, not every bike will have a corresponding entry in the SoldBike table.

I initially misunderstood the structure of the table due to an incorrect assumption that E_NUM and D_CODE together form a composite primary key. In reality, E_NUM is the only unique attribute in the table. This oversight led me to miss the presence of transitive dependencies. I should have carefully analyzed the entire table to understand its dependencies and structure more accurately. This misunderstanding caused the relational schema and the dependency diagram to be incorrect.

I think this is an improvement because I have managed to pick the little details I missed which are very important and I now know in the future I have to get clear understanding of problem given before providing solutions.

Assignment 3

Draft

Hokwana Oyintanda
Ngwenya Nomacebo Assignment3Draft 2024/04/08

Submitted individually also? Please explain.

1.

```
SELECT DISTINCT c.CUST_LNAME,c.CUST_FNAME,c.CUST_PROV,c.CUST_POSTAL
from CUSTOMER as c
JOIN INVOICE as inv on c.CUST_CODE = inv.CUST_CODE
JOIN INV_LINE as il on inv.INV_NUM=il.INV_NUM
JOIN PRODUCT as pr on il.PROD_UPC = pr.PROD_UPC
JOIN BRAND as br on pr.BRAND_ID = br.BRAND_ID
WHERE pr.PROD_CAT = 'Top Coat' AND br.BRAND_NAME = 'BUSTERS'
AND inv.INV_DATE BETWEEN '2020-06-20' AND '2020-06-25'
ORDER BY c.CUST_POSTAL desc
```

	CUST_LNAME	CUST_FNAME	CUST_PROV	CUST_POSTAL
1	MEAD	LENARD	IN	47224
2	GUERRA	ANITA	OH	44853
3	FERNANDEZ	IONE	KY	40481
4	SLAUGHTER	LEOPOLDO	MS	38753
5	CANTRELL	MONICA	AL	36865
6	ARELLANO	ERASMO	FL	32692
7	GALLAGHER	LAWRENCE	NC	27313
8	EDWARDS	VIRGIE	PA	17867
9	JEFFERS	KRISTAL	PA	17355
10	VOGT	SEAN	PA	15650
11	MEJIA	THEODORA	PA	15217
12	TALLEY	CESAR	PA	15202
13	SAMS	STEVE	NY	14081
14	ANDERSON	ERWIN	RI	02881
15	HEALY	LAWERENCE	MA	01844

# of rows? Required for ALL resultant sets.

2. Suhdsd

```
SELECT e.EMP_NUM,e.EMP_LNAME,e.EMP_EMAIL,e.EMP_TITLE
from EMPLOYEE as e
JOIN DEPARTMENT as d ON d.DEPT_NUM = e.DEPT_NUM
WHERE d.DEPT_NAME = 'CUSTOMER SERVICE' AND e.EMP_TITLE like '%ASSOCIATE%'
ORDER BY e.EMP_TITLE desc, e.EMP_LNAME
```

Hokwana Oyintanda
Ngwenya Nomacebo Assignment3Draft 2024/04/08

	EMP_NUM	EMP_LNAME	EMP_EMAIL	EMP_TITLE
1	84544	BAUER	G.BAUER88@LGCOMPANY.COM	SENIOR ASSOCIATE
2	83770	CRAIG	H.CRAIG5@LGCOMPANY.COM	SENIOR ASSOCIATE
3	83308	LOVE	C.LOVE77@LGCOMPANY.COM	SENIOR ASSOCIATE
4	83941	OBRIEN	T.OBRIEN88@LGCOMPANY.COM	SENIOR ASSOCIATE
5	83411	REDMOND	R.REDMON75@LGCOMPANY.COM	SENIOR ASSOCIATE
6	84591	SMALL	L.SMALL8@LGCOMPANY.COM	SENIOR ASSOCIATE
7	83748	WEIR	C.WEIR80@LGCOMPANY.COM	SENIOR ASSOCIATE
8	83658	MARINO	S.MARINO95@LGCOMPANY.COM	ASSOCIATE MANAGER
9	83669	RUSHING	M.RUSHIN76@LGCOMPANY.COM	ASSOCIATE MANAGER
10	84186	BARTLETT	S.BARTLE78@LGCOMPANY.COM	ASSOCIATE
11	84476	BLANKENSHIP	F.BLANKE85@LGCOMPANY.COM	ASSOCIATE
12	83824	BOUCHARD	P.BOUCHA95@LGCOMPANY.COM	ASSOCIATE
13	84265	CASH	P.CASH79@LGCOMPANY.COM	ASSOCIATE
14	84240	COLEMAN	R.COLEMA91@LGCOMPANY.COM	ASSOCIATE
15	83341	CORTEZ	C.CORTEZ85@LGCOMPANY.COM	ASSOCIATE
16	83758	DELEON	O.DELEON0@LGCOMPANY.COM	ASSOCIATE
17	83721	EASON	K.EASON99@LGCOMPANY.COM	ASSOCIATE
18	83451	ELLIS	R.ELLIS81@LGCOMPANY.COM	ASSOCIATE
19	84599	FENTON	E.FENTON3@LGCOMPANY.COM	ASSOCIATE
20	84459	GILLIAM	E.GILLIA10@LGCOMPANY.COM	ASSOCIATE
21	84442	GREGORY	A.GREGOR95@LGCOMPANY.COM	ASSOCIATE
22	84206	HEALY	N.HEALY82@LGCOMPANY.COM	ASSOCIATE
23	83728	KERNS	S.KERNS83@LGCOMPANY.COM	ASSOCIATE
24	84023	KEYES	J.KEYES10@LGCOMPANY.COM	ASSOCIATE
25	84463	LOVE	L.LOVE91@LGCOMPANY.COM	ASSOCIATE
26	84298	LYLES	J.LYLES77@LGCOMPANY.COM	ASSOCIATE
27	83534	NADEAU	B.NADEAU2@LGCOMPANY.COM	ASSOCIATE
28	84007	PETERS	L.PETERS80@LGCOMPANY.COM	ASSOCIATE
29	84287	PETTY	F.PETTY84@LGCOMPANY.COM	ASSOCIATE
30	84202	SAMUELS	C.SAMUEL84@LGCOMPANY.COM	ASSOCIATE
31	84300	SEAY	A.SEAY75@LGCOMPANY.COM	ASSOCIATE
32	83555	SPEARS	L.SPEARS82@LGCOMPANY.COM	ASSOCIATE
33	84394	STONE	P.STONE75@LGCOMPANY.COM	ASSOCIATE
34	83545	TAYLOR	J.TAYLOR86@LGCOMPANY.COM	ASSOCIATE
35	84278	TOBIN	I.TOBIN95@LGCOMPANY.COM	ASSOCIATE
36	83466	TRUJILLO	H.TRUJIL10@LGCOMPANY.COM	ASSOCIATE

3.
```sql
SELECT br.BRAND_NAME,COUNT(prod.PROD_UPC) as NR_OF_PRODUCTS
from BRAND as br
JOIN PRODUCT as prod ON br.BRAND_ID = prod.BRAND_ID
GROUP BY br.BRAND_NAME;
```

Hokwana Oyintanda
Ngwenya Nomacebo Assignment3Draft 2024/04/08

	BRAND_NAME	NR_OF_PRODUCTS
1	BINDER PRIME	27
2	BUSTERS	25
3	FORESTERS BEST	15
4	HOME COMFORT	36
5	LE MODE	36
6	LONG HAUL	41
7	OLDE TYME QUALITY	27
8	STUTTENFURST	27
9	VALU-MATTE	18

4.

```
ELECT emp.EMP_NUM,CONCAT(emp.EMP_LNAME, ', ', emp.EMP_FNAME) AS
EMP_NAME,sh.SAL_FROM,sh.SAL_AMOUNT
from EMPLOYEE as emp
JOIN SALARY_HISTORY as sh ON emp.EMP_NUM = sh.EMP_NUM
WHERE sh.SAL_END  IS NULL
ORDER BY sh.SAL_FROM desc
```

	EMP_NUM	EMP_NAME	SAL_FROM	SAL_AMOUNT
1	84047	HOPPER, EMMITT	2021-01-30 00:00:00.000	34250.00
2	83624	ELLIOT, GERTRUDE	2016-01-29 00:00:00.000	70560.00
3	83542	LONGORIA, TOSHA	2016-01-29 00:00:00.000	32000.00
4	84594	TIDWELL, ODELL	2016-01-28 00:00:00.000	77400.00
5	83669	RUSHING, MITCHELL	2016-01-28 00:00:00.000	32200.00
6	84017	WALDRON, IGNACIO	2016-01-27 00:00:00.000	30000.00
7	84328	CARPENTER, FERN	2016-01-27 00:00:00.000	94090.00
8	83603	PADGETT, TANIKA	2016-01-27 00:00:00.000	33250.00
9	83838	FLORES, LEN	2016-01-25 00:00:00.000	58880.00
10	83914	DUKE, REYNA	2016-01-24 00:00:00.000	45120.00
11	83609	GIBBS, SHAUN	2016-01-23 00:00:00.000	89550.00
12	84502	FITZPATRICK, SAL	2016-01-20 00:00:00.000	73140.00
13	84333	CONKLIN, PHILLIS	2016-01-19 00:00:00.000	62100.00
14	84233	WELCH, ALIDA	2016-01-18 00:00:00.000	29500.00
15	83537	ENGLISH, CLEO	2016-01-16 00:00:00.000	136000.00
16	83763	FELTON, JAIME	2016-01-12 00:00:00.000	107000.00
17	83964	SWEENEY, HAILEY	2016-01-12 00:00:00.000	85000.00
18	83630	DOVE, BRENDAN	2016-01-11 00:00:00.000	47000.00
19	84023	KEYES, JAROD	2016-01-10 00:00:00.000	44250.00
20	84123	PLATT, MARGOT	2016-01-10 00:00:00.000	31000.00
21	84219	WILKINSON, THURMAN	2016-01-08 00:00:00.000	71000.00
22	84256	DODGE, EDGAR	2016-01-08 00:00:00.000	47500.00
23	83564	CONWAY, EDGARDO	2016-01-08 00:00:00.000	26500.00

5.

```
SELECT emp.DEPT_NUM,MAX(emp.EMP_HIREDATE) AS LastHired
FROM EMPLOYEE AS emp
GROUP BY emp.DEPT_NUM
ORDER BY emp.DEPT_NUM ASC
```

Hokwana Oyintanda
Ngwenya Nomacebo Assignment3Draft 2024/04/08

	DEPT_NUM	LastHired
1	200	2005-06-08 00:00:00.000
2	250	2015-12-15 00:00:00.000
3	280	2014-04-16 00:00:00.000
4	300	2014-12-12 00:00:00.000
5	400	2015-01-26 00:00:00.000
6	500	2015-04-26 00:00:00.000
7	550	2015-10-22 00:00:00.000
8	600	2015-10-02 00:00:00.000

6.

```
SELECT cu.CUST_CODE,cu.CUST_LNAME,SUM(inv.INV_TOTAL) AS Total_Invoice
FROM CUSTOMER as cu
JOIN INVOICE as inv ON cu.CUST_CODE = inv.CUST_CODE
WHERE cu.CUST_CODE IN ( SELECT CUST_CODE
        FROM (SELECT CUST_CODE,SUM(INV_TOTAL) AS Cumulative_Total
             FROM INVOICE
             GROUP BY CUST_CODE
             HAVING SUM(INV_TOTAL) >= 3000) AS Cumulative)
GROUP BY cu.CUST_CODE, cu.CUST_LNAME
ORDER BY Total_Invoice ASC;
```

Do this without a sub-query.

Resultant table?

Final

Hokwana Oyintanda 27/27 Assignment3Final 2024/04/15

1.

```
SELECT DISTINCT c.CUST_LNAME,c.CUST_FNAME,c.CUST_PROV,c.CUST_POSTAL
from CUSTOMER as c
JOIN INVOICE as inv on c.CUST_CODE = inv.CUST_CODE
JOIN INV_LINE as il on inv.INV_NUM=il.INV_NUM
JOIN PRODUCT as pr on il.PROD_UPC = pr.PROD_UPC
JOIN BRAND as br on pr.BRAND_ID = br.BRAND_ID
WHERE pr.PROD_CAT = 'Top Coat' AND br.BRAND_NAME = 'BUSTERS'
AND inv.INV_DATE BETWEEN '2020-06-20' AND '2020-06-25'
ORDER BY c.CUST_POSTAL desc
```

[7] ✓

	CUST_LNAME	CUST_FNAME	CUST_PROV	CUST_POSTAL
1	MEAD	LENARD	IN	47224
2	GUERRA	ANITA	OH	44853
3	FERNANDEZ	IONE	KY	40481
4	SLAUGHTER	LEOPOLDO	MS	38753
5	CANTRELL	MONICA	AL	36865
6	ARELLANO	ERASMO	FL	32692
7	GALLAGHER	LAWRENCE	NC	27313
8	EDWARDS	VIRGIE	PA	17867
9	JEFFERS	KRISTAL	PA	17355
10	VOGT	SEAN	PA	15650
11	MEJIA	THEODORA	PA	15217
12	TALLEY	CESAR	PA	15202
13	SAMS	STEVE	NY	14081
14	ANDERSON	ERWIN	RI	02881
15	HEALY	LAWERENCE	MA	01844

SAM (16.0 RTM) | SAM\Samkele Hokwana (63) | Hardware2024 | 00:00:00 | 15 rows

2.

```
SELECT e.EMP_NUM,e.EMP_LNAME,e.EMP_EMAIL,e.EMP_TITLE
from EMPLOYEE as e
JOIN DEPARTMENT as d ON d.DEPT_NUM = e.DEPT_NUM
WHERE d.DEPT_NAME = 'CUSTOMER SERVICE' AND e.EMP_TITLE like '%ASSOCIATE%'
ORDER BY e.EMP_TITLE desc, e.EMP_LNAME
```

[4] ✓

Hokwana Oyintanda Assignment3Final 2024/04/15

	EMP_NUM	EMP_LNAME	EMP_EMAIL	EMP_TITLE
1	84544	BAUER	G.BAUER88@LGCOMPANY.COM	SENIOR ASSOCIATE
2	83770	CRAIG	H.CRAIG5@LGCOMPANY.COM	SENIOR ASSOCIATE
3	83308	LOVE	C.LOVE77@LGCOMPANY.COM	SENIOR ASSOCIATE
4	83941	OBRIEN	T.OBRIEN88@LGCOMPANY.COM	SENIOR ASSOCIATE
5	83411	REDMOND	R.REDMON75@LGCOMPANY.COM	SENIOR ASSOCIATE
6	84591	SMALL	L.SMALL8@LGCOMPANY.COM	SENIOR ASSOCIATE
7	83748	WEIR	C.WEIR80@LGCOMPANY.COM	SENIOR ASSOCIATE
8	83658	MARINO	S.MARINO95@LGCOMPANY.COM	ASSOCIATE MANAGER
9	83669	RUSHING	M.RUSHIN76@LGCOMPANY.COM	ASSOCIATE MANAGER
10	84186	BARTLETT	S.BARTLE78@LGCOMPANY.COM	ASSOCIATE
11	84476	BLANKENSHIP	F.BLANKE85@LGCOMPANY.COM	ASSOCIATE
12	83824	BOUCHARD	P.BOUCHA95@LGCOMPANY.COM	ASSOCIATE
13	84265	CASH	P.CASH79@LGCOMPANY.COM	ASSOCIATE
14	84240	COLEMAN	R.COLEMA91@LGCOMPANY.COM	ASSOCIATE
15	83341	CORTEZ	C.CORTEZ85@LGCOMPANY.COM	ASSOCIATE
16	83758	DELEON	O.DELEON0@LGCOMPANY.COM	ASSOCIATE
17	83721	EASON	K.EASON99@LGCOMPANY.COM	ASSOCIATE
18	83451	ELLIS	R.ELLIS81@LGCOMPANY.COM	ASSOCIATE
19	84599	FENTON	E.FENTON3@LGCOMPANY.COM	ASSOCIATE
20	84459	GILLIAM	E.GILLIA10@LGCOMPANY.COM	ASSOCIATE
21	84442	GREGORY	A.GREGOR95@LGCOMPANY.COM	ASSOCIATE
22	84206	HEALY	N.HEALY82@LGCOMPANY.COM	ASSOCIATE
23	83728	KERNS	S.KERNS83@LGCOMPANY.COM	ASSOCIATE
24	84023	KEYES	J.KEYES10@LGCOMPANY.COM	ASSOCIATE
25	84463	LOVE	L.LOVE91@LGCOMPANY.COM	ASSOCIATE
26	84298	LYLES	J.LYLES77@LGCOMPANY.COM	ASSOCIATE
27	83534	NADEAU	B.NADEAU2@LGCOMPANY.COM	ASSOCIATE
28	84007	PETERS	L.PETERS80@LGCOMPANY.COM	ASSOCIATE
29	84287	PETTY	F.PETTY84@LGCOMPANY.COM	ASSOCIATE
30	84202	SAMUELS	C.SAMUEL84@LGCOMPANY.COM	ASSOCIATE
31	84300	SEAY	A.SEAY75@LGCOMPANY.COM	ASSOCIATE
32	83555	SPEARS	L.SPEARS82@LGCOMPANY.COM	ASSOCIATE
33	84394	STONE	P.STONE75@LGCOMPANY.COM	ASSOCIATE
34	83545	TAYLOR	J.TAYLOR86@LGCOMPANY.COM	ASSOCIATE
35	84278	TOBIN	I.TOBIN95@LGCOMPANY.COM	ASSOCIATE
36	83466	TRUJILLO	H.TRUJIL10@LGCOMPANY.COM	ASSOCIATE

SAM (16.0 RTM) | SAM\Samkele Hokwana (62) | Hardware2024 | 00:00:00 | 38 rows

3.

```
SELECT br.BRAND_NAME,COUNT(prod.PROD_UPC) as NR_OF_PRODUCTS
from BRAND as br
JOIN PRODUCT as prod ON br.BRAND_ID = prod.BRAND_ID
GROUP BY br.BRAND_NAME;
```

(3) ✓

Hokwana Oyintanda Assignment3Final 2024/04/15

	BRAND_NAME	NR_OF_PRODUCTS
1	BINDER PRIME	27
2	BUSTERS	25
3	FORESTERS BEST	15
4	HOME COMFORT	36
5	LE MODE	36
6	LONG HAUL	41
7	OLDE TYME QUALITY	27
8	STUTTENFURST	27
9	VALU-MATTE	18

SAM (16.0 RTM) | SAM\Samkele Hokwana (61) | Hardware2024 | 00:00:00 | 9 rows

4.

```
ELECT emp.EMP_NUM,CONCAT(emp.EMP_LNAME, ', ', emp.EMP_FNAME) AS
EMP_NAME,sh.SAL_FROM,sh.SAL_AMOUNT
from EMPLOYEE as emp
JOIN SALARY_HISTORY as sh ON emp.EMP_NUM = sh.EMP_NUM
WHERE sh.SAL_END  IS NULL
ORDER BY sh.SAL_FROM desc
```

(5) ✓

	EMP_NUM	EMP_NAME	SAL_FROM	SAL_AMOUNT
1	84047	HOPPER, EMMITT	2021-01-30 00:00:00.000	34250.00
2	83624	ELLIOT, GERTRUDE	2016-01-29 00:00:00.000	70560.00
3	83542	LONGORIA, TOSHA	2016-01-29 00:00:00.000	32000.00
4	84594	TIDWELL, ODELL	2016-01-28 00:00:00.000	77400.00
5	83669	RUSHING, MITCHELL	2016-01-28 00:00:00.000	32200.00
6	84017	WALDRON, IGNACIO	2016-01-27 00:00:00.000	30000.00
7	84328	CARPENTER, FERN	2016-01-27 00:00:00.000	94090.00
8	83603	PADGETT, TANIKA	2016-01-27 00:00:00.000	33250.00
9	83838	FLORES, LEN	2016-01-25 00:00:00.000	58880.00
10	83914	DUKE, REYNA	2016-01-24 00:00:00.000	45120.00
11	83609	GIBBS, SHAUN	2016-01-23 00:00:00.000	89550.00
12	84502	FITZPATRICK, SAL	2016-01-20 00:00:00.000	73140.00
13	84333	CONKLIN, PHILLIS	2016-01-19 00:00:00.000	62100.00
14	84233	WELCH, ALIDA	2016-01-18 00:00:00.000	29500.00
15	83537	ENGLISH, CLEO	2016-01-16 00:00:00.000	136000.00
16	83763	FELTON, JAIME	2016-01-12 00:00:00.000	107000.00
17	83964	SWEENEY, HAILEY	2016-01-12 00:00:00.000	85000.00
18	83630	DOVE, BRENDAN	2016-01-11 00:00:00.000	47000.00
19	84023	KEYES, JAROD	2016-01-10 00:00:00.000	44250.00
20	84123	PLATT, MARGOT	2016-01-10 00:00:00.000	31000.00
21	84219	WILKINSON, THURMAN	2016-01-08 00:00:00.000	71000.00
22	84256	DODGE, EDGAR	2016-01-08 00:00:00.000	47500.00
23	83564	CONWAY, EDGARDO	2016-01-08 00:00:00.000	26500.00

SAM (16.0 RTM) | SAM\Samkele Hokwana (52) | Hardware2024 | 00:00:00 | 363 rows

5.

Hokwana Oyintanda Assignment3Final 2024/04/15

(3)

```
SELECT emp.DEPT_NUM,MAX(emp.EMP_HIREDATE) AS LastHired
FROM EMPLOYEE AS emp
GROUP BY emp.DEPT_NUM
ORDER BY emp.DEPT_NUM ASC
```

	DEPT_NUM	LastHired
1	200	2005-06-08 00:00:00.000
2	250	2015-12-15 00:00:00.000
3	280	2014-04-16 00:00:00.000
4	300	2014-12-12 00:00:00.000
5	400	2015-01-26 00:00:00.000
6	500	2015-04-26 00:00:00.000
7	550	2015-10-22 00:00:00.000
8	600	2015-10-02 00:00:00.000

SAM (16.0 RTM) | SAM\Samkele Hokwana (57) | Hardware2024 | 00:00:00 | 8 rows

6.

(5)

```
SELECT cu.CUST_CODE,cu.CUST_LNAME,SUM(inv.INV_TOTAL) AS Total_Invoice
FROM CUSTOMER as cu
JOIN INVOICE as inv ON cu.CUST_CODE = inv.CUST_CODE
GROUP BY cu.CUST_CODE, cu.CUST_LNAME
HAVING SUM(inv.INV_TOTAL) >= 3000
ORDER BY Total_Invoice ASC;
```

	CUST_CODE	CUST_LNAME	Total_Invoice
1	117	MATA	3009.63
2	152	WHITTAKER	3042.78
3	98	MARINO	3052.46
4	215	BRYAN	3134.15

SAM (16.0 RTM) | SAM\Samkele Hokwana (56) | Hardware2024 | 00:00:00 | 4 rows

Assignment 4

Draft

Hokwana Oyintanda Samkele
Masindi Muedi Assignment4Draft 2024/04/22

1.

a)
Top-down strategy starts with recognizing the different elements of the design and then combines them into more substantial groupings. Defining attributes is the first step in database architecture, after which they are grouped into entities. In contrast to top-down design. We use this approach in this module when we were creating schemas, we had to start by defining the attributes and then group into entities. We have used Top-down in ER Modelling.

no.

b)
Bottom-up strategy divides the data elements (items) into data sets after first identifying the individual items. Stated differently, it defines properties first, then groups them together to create entities. Whenever we talk about the process of creating a database schema, we frequently begin by figuring out which specific data components or attributes must be kept. These characteristics, which are the smallest data units in the database, are specified in accordance with the particular data that must be recorded for every entity. We have used bottom up in Normalisation.

2.

a) Only implement the supertype as a table meaning in this case implement Employee table. Attributes of subtype (Nurse and Doctor) become attributes of supertype (Employee) meaning *position* and *shiftHours* become attributes of Employee *null* when not used. empType is the subtype discriminator to indicate if the row represents Doctor, or Nurse, or any other type of Employee and empNo is Primary key.

Only implement subtypes as tables meaning only implement Doctor and Nurse tables. Attributes of supertype (Employee) are held in all subtypes (Doctor and Nurse) meaning both Doctor and Nurse tables will contain all attributes of Employee table. empNo is PK and FK in both Doctor and Nurse tables.

Why? Are there any others in the given model?

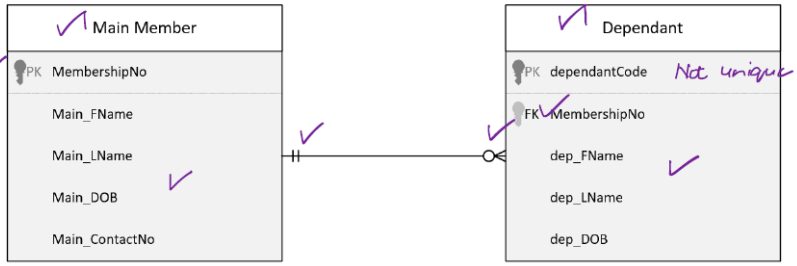
Supertype and Subtype should be separate tables meaning Employee, Doctor and Nurse are different tables. Employee table will contain all its attributes and empType as subtype discriminator to distinguish between Doctor, Nurse, and other Employees. empNo is PK and FK in both Doctor and Nurse tables. Doctor table also have *position* attribute. Nurse table also have *shiftHours* attribute.

Hokwana Oyintanda Samkele
Masindi Muedi Assignment4Draft 2024/04/22

b) Supertype and Subtype should be separate tables. This is suited because the clinic has other Employees that are not Doctor or Nurse, it may be cleaners or securities.

3.

a)



b) MainMember (MembershipNo[PK],Main_FName,Main_LName,Main_DOB,Main_ContactNo)
Dependant(dependantCode[PK],MembershipNo[FK],dep_FName, dep_LName,dep_DOB)

4.

a)

```
BEGIN TRANSACTION

UPDATE MedContent
SET MC_QOH = MC_QOH - (250 * 30000)
WHERE MC_CODE = 'PCM'
UPDATE MedContent
SET MC_QOH = MC_QOH - (200 * 30000)
WHERE MC_CODE = 'IP'
UPDATE MedContent
SET MC_QOH = MC_QOH - (10 * 30000)
WHERE MC_CODE = 'CP'
UPDATE Medication
```

Hokwana Oyintanda Samkele
Masindi Muedi Assignment4Draft 2024/04/22

```
SET MEDS_QOH = MEDS_QOH + 30000
WHERE MEDS_CODE = 'MypCap'
COMMIT
```

TRL ID	TRX NUM	PREV PTR	NXT PTR	OPERATION	TABLE	ROW ID	ATTRIBUTE	BEFORE VALUE	AFTER VALUE
91	13	NULL	92	START	*START TRANS*				
92	13	91	93	UPDATE	COMPONENT	PCM	MC_QOH	1990000000	1 982 500 000
93	13	92	94	UPDATE	COMPONENT	IP	MC_QOH	450 400000	444 400 000
94	13	93	95	UPDATE	COMPONENT	CP	MC_QOH	753005000	75 000 500
95	13	94	96	UPDATE	PRODUCT	MypCap	MEDS_QOH	740300	770 300
96	13	95	NULL	COMMIT	*END TRANS*				

Final

Hokwana Oyintanda Samkele
Masindi Muedi Assignment4Final 2024/04/29

41/45

1.

a)
Top-down strategy start with higher level entity supertype, then identify lower level, more specific entity types (subtypes). Group unique characteristics and relationships of subtypes. It also allows for attribute inheritance. It defines subtype discriminator (in supertype). We have used Top-down in ER Modelling. At what stage are you defining attributes? Before or after defining entities? -1

[3]

b)
Bottom-up strategy divides the data elements (items) into data sets after first identifying the individual items. Stated differently, it defines properties first, then groups them together to create entities. Whenever we talk about the process of creating a database schema, we frequently begin by figuring out which specific data components or attributes must be kept. These characteristics, which are the smallest data units in the database, are specified in accordance with the particular data that must be recorded for every entity. We have used bottom up in Normalisation. ✓

2.

[10]

a) Only implement the supertype as a table meaning in this case implement Employee table. Attributes of subtype (Nurse and Doctor) become attributes of supertype (Employee) meaning *position* and *shiftHours* become attributes of Employee *null* when not used. empType is the subtype discriminator to indicate if the row represents Doctor, or Nurse, or any other type of Employee and empNo is Primary key. are there any other types? -1

7

Only implement subtypes as tables meaning only implement Doctor and Nurse tables. Attributes of supertype (Employee) are held in all subtypes (Doctor and Nurse) meaning both Doctor and Nurse tables will contain all attributes of Employee table. empNo is PK in both Doctor and Nurse tables. any other attributes? -1

Supertype and Subtype should be separate tables meaning Employee, Doctor and Nurse are different tables. Employee table will contain all its attributes and empType as subtype discriminator to distinguish between Doctor, Nurse. empNo is PK and FK in both Doctor and Nurse tables. Doctor table also have *position* attribute. Nurse table also have *shiftHours* attribute. ✓

3

b) Only implement subtypes as tables meaning only implement Doctor and Nurse tables. This is suited because an Employee is either Doctor OR Nurse and Employee is abstract. What does this mean? -1 ✓

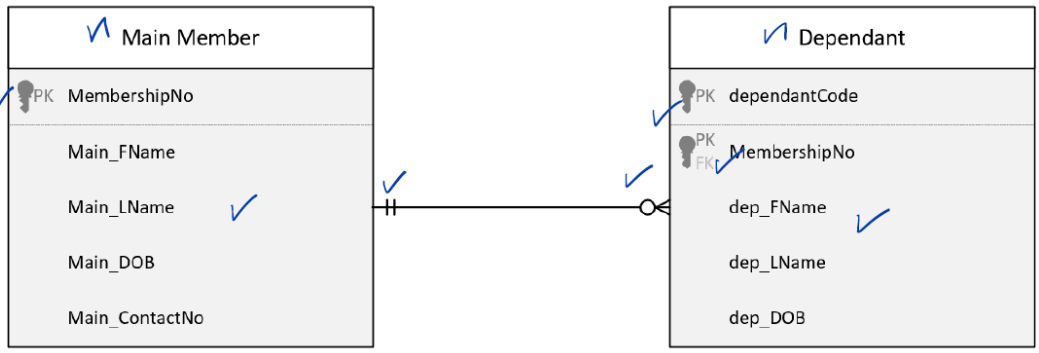
Contradictory

Hokwana Oyintanda Samkele
Masindi Muedi Assignment4Final 2024/04/29

3.

[13]

a)



8

5

b) MainMember (MembershipNo[PK],Main_FName,Main_LName,Main_DOB,Main_ContactNo) ✓
Dependant(dependantCode[PK],MembershipNo[FK][PK],dep_FName, dep_LName,dep_DOB) ✓

4.

[15]

a)

```
BEGIN TRANSACTION

UPDATE MedContent
SET MC_QOH = MC_QOH - (250 * 30000)
WHERE MC_CODE = 'PCM'
UPDATE MedContent
SET MC_QOH = MC_QOH - (200 * 30000)
WHERE MC_CODE = 'IP'
```

6

Hokwana Oyintanda Samkele
Masindi Muedi Assignment4Final 2024/04/29

```
UPDATE MedContent
SET MC_QOH = MC_QOH - (10 * 30000)
WHERE MC_CODE = 'CP'
UPDATE Medication
SET MEDS_QOH = MEDS_QOH + 30000
WHERE MEDS_CODE = 'MypCap'
COMMIT
```

b)

TRL ID	TRX NUM	PREV PTR	NXT PTR	OPERATION	TABLE	ROW ID	ATTRIBUTE	BEFORE VALUE	AFTER VALUE
91	13	NULL	92	START	*START TRANS*				
92	13	91	93	UPDATE	MedContent	PCM	MC_QOH	1990000000	1 982 500 000
93	13	92	94	UPDATE	MedContent	IP	MC_QOH	450 400000	444 400 000
94	13	93	95	UPDATE	MedContent	CP	MC_QOH	753005000	75 000 500
95	13	94	96	UPDATE	Medication	MypCap	MEDS_QOH	740300	770 300
96	13	95	NULL	COMMIT	*END TRANS*				

Assignment 5

Draft

Hokwana Oyintanda Assignment5Draft 2024/05/06

1.
   a) High. We have larger number of distinct values that can be inserted into this column.
   b) Create an Index on areaCode because it appears on WHERE clause and the data sparsity on areaCode is high. Create Index on empLName and empFName because they appear on ORDER BY clause.
   CREATE INDEX idx_empAreaCode ON Employee (empAreaCode);
   CREATE INDEX idx_empName ON Employee (empLName, empFName);

2.
   a)
```
SELECT A.agentLName, A.agentFName, L.listID, L.Suburb, S.SalePrice
FROM SALE AS S
INNER JOIN AGENT AS A ON S.agentID = A.agentID
INNER JOIN LIST AS L ON L.listID = S.listID
WHERE L.Suburb = 'Summerstrand'
  AND S.SalePrice <= 2500000
  AND S.SaleDate > '2024-02-01'
ORDER BY S.SaleDate;
```

   Remove the old-style joins, use inner join instead.
   Move equality conditions earlier (faster than inequality).
   Got rid of NOT condition because slower to execute and use inequality instead.
   Then check for date, since are slower to execute.

   b) Given the query's filtering conditions and join operations, indexing SaleDate and SalePrice in the SALE table, and Suburb in the LIST table would likely yield the most significant performance improvements. These indexes would facilitate faster retrieval of relevant rows based on the filtering criteria, thereby enhancing the overall query execution speed.

   Additionally, indexing agentID in both the SALE and AGENT tables, as well as listID in both the SALE and LIST tables, would optimize the performance of the join operations. This improvement arises from the ability to rapidly locate related rows in the associated tables, thereby streamlining the join process and reducing query execution time.

3. See slide 19 of Chapter 14 slides. Remote / distributed transaction / request?
   a) SELECT: The database must support the SELECT operation to query data from the SUPPLIER table.
   FILTERING: It needs to support filtering capabilities to specify the condition where SUP_TOT < 35000.
   b) BEGIN TRANSACTION
   UPDATE SUPPLIER
   INSERT INTO ORDER
   INSERT INTO ORDER_LINE
   UPDATE STOCK
   COMMIT

Final

Hokwana Oyintanda Assignment5Final 2024/05/13

1.

a) High. We have larger number of distinct values that can be inserted into this column

b) Create an Index on areaCode because it appears on WHERE clause and the data sparsity on areaCode is high. Create Index on empLName and empFName because they appear on ORDER BY clause.

```
CREATE INDEX idx_empAreaCode ON Employee (empAreaCode);
CREATE INDEX idx_empName ON Employee (empLName, empFName);
```

2.

a)
```
SELECT A.agentLName, A.agentFName, L.listID, L.Suburb, S.SalePrice
FROM SALE AS S
INNER JOIN AGENT AS A ON S.agentID = A.agentID
INNER JOIN LIST AS L ON L.listID = S.listID
WHERE L.Suburb = 'Summerstrand'
  AND S.SalePrice <= 2500000
  AND S.SaleDate > '2024-02-01'
ORDER BY S.SaleDate;
```

Remove the old-style joins, use inner join instead.
Move equality conditions earlier (faster than inequality).
Got rid of NOT condition because slower to execute and use inequality instead.
Then check for date, since are slower to execute.

b) Given the query's filtering conditions and join operations, indexing SaleDate and SalePrice in the SALE table, and Suburb in the LIST table would likely yield the most significant performance improvements. These indexes would facilitate faster retrieval of relevant rows based on the filtering criteria, thereby enhancing the overall query execution speed.

3.

a) SELECT SUPPLIER

- Distributed request. We have single SQL statement (SELECT) referencing data located at several local or remote sites (Site A and Site C)

Entire transaction

- Distributed request. The entire transaction is distributed request because we have single request which is referencing data located in several sites i.e Site A and Site C.

b) UPDATE SUPPLIER

- Distributed request. We have single SQL statement (UPDATE) referencing data located at several local or remote sites (Site A and Site C)

INSERT INTO ORDER

- Remote request. We have Single SQL statement (INSERT) accesses data at single site (Site B).

INSERT INTO ORDER_LINE

- Remote request. We have a single request referencing data located single site i.e Site B.

Reflection

The most valuable concept in this module is the process of identifying business rules and then determining the relationships based on those rules. Additionally, the process of normalization, which involves refining table structures of an already existing database, is crucial. This dual focus not only ensures a clear understanding of the underlying principles governing data but also enhances the efficiency and integrity of the database design. I hope with this understanding, I will grasp database concepts thoroughly and apply them effectively in the workplace.

One of the aspects of this module that has been done particularly well is the alignment of assignments with both the material covered in class and the upcoming week's content. This approach allows students to attempt work that hasn't yet been taught, fostering independent learning and exploration. By engaging with new concepts on their own first, students can then attend class already familiar with the topics, using the opportunity to clarify and deepen their understanding. This method enhances the learning experience by encouraging proactive engagement with the material.

Honestly, I don't think there is any need for improvement. The only thing required is to maintain this high level of competency.

One life lesson I can take away from this module is the value of learning independently and then applying that knowledge in group settings. Initially, I faced the challenge of explaining my understanding to others during group assignments. However, I overcame this challenge and now feel confident in articulating my opinions to others. Additionally, I have gained a better understanding of how people view things differently. These group assignments have been incredibly valuable for developing both my communication and collaboration skills.