**CSCI235: Database Systems**

**Final Exam**

**Nicholas Monteleone 5055076**

**Question 1**

1. A → B; B → AD; transitive A → AD

Therefore A → BD

Minimal Keys: AC, BC

Schema does not satisfy 2NF because:

B → AD and B is a subset of BC

A → B and A is a subset of AC

Therefore schema is 1NF and does not satisfy any higher order Normal Forms

R(B,A,D)

A → B; B → AD; transitive A → AD

Minimal Keys: A, B

R(B,C)

Minimal Keys: BC

All schemas are in BCNF

1. Minimal Keys: BCD

Schema does not satisfy 2NF because:

B → A and B is a subset of BCD

Therefore schema is 1NF and does not satisfy any higher order Normal Forms

R(B,A)

B → A

Minimal Keys: B

R(B,C,D)

Minimal Keys: BCD

All schemas are in BCNF

1. D → A; A → BCD; transitive D → BCD

Therefore D → ABC

Minimal Keys: D

Schema satisfies 2NF as all attributes are fully dependant on primary keys

Schema satisfies 3NF as there are no transitive functional dependencies

Schema satisfies BCNF as all minimal keys are super keys

1. Minimal Keys: ABCD

Schema satisfies 2NF as all attributes are fully dependant on primary keys

Schema satisfies 3NF as there are no transitive functional dependencies

Schema satisfies BCNF as all minimal keys are super keys

**Question 2**

**Question 3**

|  |  |
| --- | --- |
| Transaction 1 | Transaction 2 |
| SELECT COUNT(\*)  INTO total\_suppliers  FROM SUPPLIER  WHERE COUNTRY = supplier\_country; |  |
|  | UPDATE SUPPLIER  SET COUNTRY = ‘An updated country name’  WHERE COUNTRY = supplier\_country; |
|  | COMMIT; |
| SELECT COUNT(\*)  INTO total\_products  FROM PRODUCT JOIN SUPPLIER  ON PRODUCT.SUPPLIER\_NAME = SUPPLIER.COMPANY\_NAME  WHERE COUNTRY = supplier.country; |  |
| IF total\_suppliers = 0 THEN  RETURN 0;  ELSE  RETURN total\_products/total\_suppliers;  END IF; |  |
| COMMIT; |  |

1. UPDATE statement processed by T2 changed all values in SUPPLIER of the country’s name. Therefore once the COMMIT is processed, no SUPPLIERs are left with the same name as when the COUNT was performed by T1.

As the transactions are processed at READ COMMITTED isolation level, the modification is read by the second COUNT of T1, and an inaccurate reading is given where total\_products will return a value not corresponding to total\_suppliers.

**Question 4**

1. Run the same SQL script ‘dbcreate.sql’ that was used on the “host server” on the “remote server” to create empty relational tables that correspond to the “host server” database
2. CREATE DATABASE LINK "DB.DATA-PC02" CONNECT TO nm824 IDENTIFIED BY zeb1b4 USING 'data-pc02.adeis.uow.edu.au:1521/db';

CREATE DATABASE LINK "DB.DATA-PC02" CONNECT TO nm824 IDENTIFIED BY zeb1b4 USING 'data-pc02.adeis.uow.edu.au:1521/db';

CREATE SYNONYM remoteCUSTOMER FOR CUSTOMER@"DB.DATA-PC02";

CREATE SYNONYM remoteORDERS FOR ORDERS@"DB.DATA-PC02";

CREATE SYNONYM remoteSUPPLIER FOR SUPPLIER@"DB.DATA-PC02";

CREATE SYNONYM remotePRODUCT FOR PRODUCT@"DB.DATA-PC02";

CREATE SYNONYM remoteORDER\_DETAIL FOR ORDER\_DETAIL@"DB.DATA-PC02";

INSERT INTO remoteCUSTOMER

SELECT \*

FROM CUSTOMER

;

INSERT INTO remoteORDERS

SELECT \*

FROM ORDERS

WHERE EXTRACT(YEAR FROM ORDER\_DATE) = 2021

;

INSERT INTO remoteSUPPLIER

SELECT \*

FROM SUPPLIER

;

INSERT INTO remotePRODUCT

SELECT \*

FROM PRODUCT

;

INSERT INTO remoteORDER\_DETAIL

SELECT ORDER\_NUM, PRODUCT\_NAME, QUANTITY

FROM ORDER\_DETAIL

WHERE ORDER\_NUM IN ( SELECT ORDER\_NUM FROM remoteORDERS)

;

DELETE FROM ORDER\_DETAIL

WHERE ORDER\_NUM IN (SELECT ORDER\_NUM FROM remoteORDER\_DETAIL)

;

DELETE FROM ORDERS

WHERE ORDER\_NUM IN (SELECT ORDER\_NUM FROM remoteORDERS)

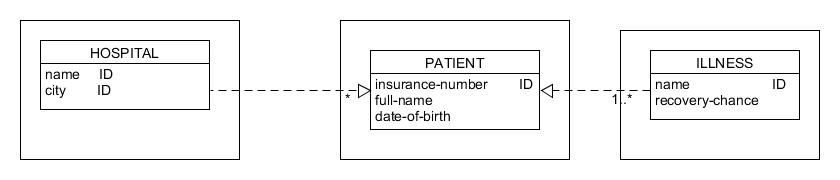
;

DELETE FROM CUSTOMERS

WHERE CUSTOMER\_CODE IN (SELECT CUSTOMER\_CODE FROM remoteORDERS)

;

**Question 5**

1. 
2. {“\_id”:”1”,

“HOSPITAL”:{“name”:”Sacred Heart”,

“city”:”Los Angeles”,

“admits”:[ {“PATIENT”: {“insurance-number”:1234,

“full-name”:”Bill Johnson”,

“date-of-birth”:Date(“1996-09-30”),

“diagnosed”:[

{“ILLNESS”:{“name”:”Crohn’s Disease”,

“recovery-chance”:0}},

{“ILLNESS”:{“name”:”Cancer”,

“recovery-chance”:0.5}}

]

}, {“PATIENT”: {“insurance-number”:1234,

“full-name”:”John Billson”,

“date-of-birth”:Date(“1990-10-22”),

“diagnosed”:[

{“ILLNESS”:{“name”:”Broken Wrist”,

“recovery-chance”:0.95}},

{“ILLNESS”:{“name”:”Broken Rib”,

“recovery-chance”:0.75}},

{“ILLNESS”:{“name”:”Broken Leg”,

“recovery-chance”:0.75}}

]

}]

}

}

**Question 6**

1. db.driver.aggregate( [ {$project:{“address.country”:1,”address.city”:1,”\_id”:0}}]);
2. db.driver.aggregate( [ {$and:{ $match:{“first\_name”:”James” },

$match:{“last\_name”:”Bond” }

}},

$project:{“trips.truck\_rego”:1,”\_id”:0}] );

1. db.driver.aggregate( [ {$project:{“first\_name”:1,”last\_name”:1,”\_id”:0}},

{$count: “total number of trips”}]);

1. db.driver.update({“first\_name”:”James”,”last\_name”:”Bond”,”trips.date”:5},

{$set:{"trips.date”:”28-SEP-2021”}});

1. db.driver.update({“first\_name”:”James”,”last\_name”:”Bond”,”trips.date”:25},

{$unset:{"trips.number”:””,”trips.truck\_rego”:””,”trips.date”:””,”trips.legs”:””}});