INFO20003 Exam - Semester 1

Suggested Solutions and Marking Scheme

Oscar Correa and Colton Carner

July 2020

0.1 ER Modelling I (Exam Q8 - 15 marks)

0.1.1 VicRoads

VicRoads wants to upgrade its database and it hired you to be in charge of the design of the roadworthiness certificates section. The relevant information is as follows:

- Every vehicle has a registration number, and each vehicle is of a specific category.
- VicRoads consider several categories, and each category is identified by code (e.g., LV for light vehicles), maximum capacity, and average weight.
- Several certified mechanics can issue roadworthiness certificates. You need to store the name, driver's license, address, phone number, and cost per hour of each mechanic.
- Each mechanic is an expert on one or more vehicle category(ies), and his or her expertise may overlap with that of other mechanics. This information about mechanics must also be recorded.
- All mechanics belong to a union. You must store the union membership number of each mechanic. You can assume that each mechanic is uniquely identified by her/his driver's license as every mechanic needs one to test the vehicle on the road.
- VicRoads has several tests that are required whenever a vehicle is to be sold, re-registered, or to clear some defect notices. Each test has a code (e.g., W&T) and a name (e.g., wheels and tyres), and a minimum score.
- VicRoads requires to store the issued certificates (one per vehicle). Each certificate comprises a set of tests by a given mechanic. The information needed for each certificate is the date, the number of hours the mechanic spent doing the test, and the score the vehicle received on each test.

Draw a conceptual model using Chen's notation for the roadworthiness tests section. Be sure to indicate the various attributes of each entity and relationship set; also specify the key and participation constraints for each relationship set. Be sure to write down any assumptions you make.

Solution

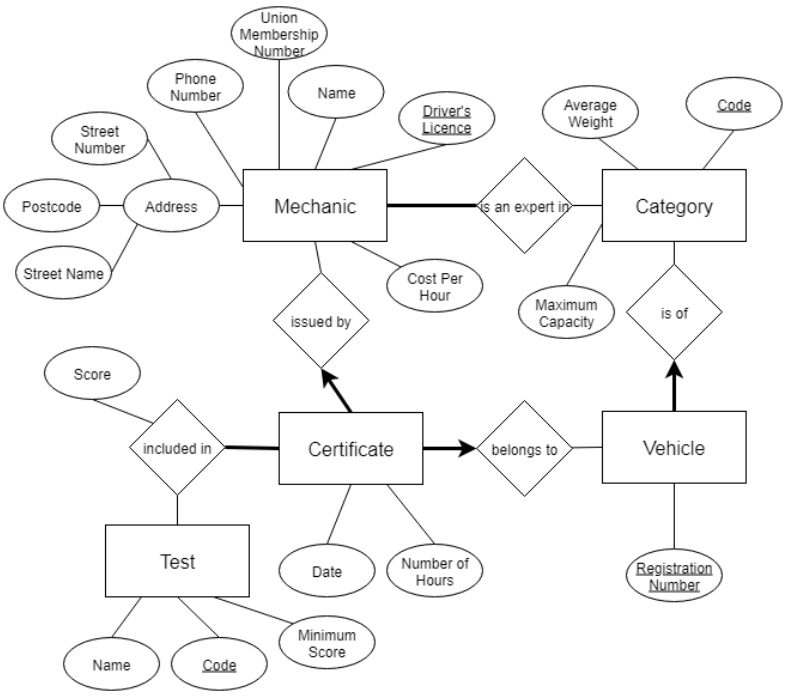


Figure 1: VicRoads Conceptual Model

0.1.2 Record Label

A record label company wants to upgrade its albums database and it hires you to do the job. The relevant information is as follows:

- They want to keep information about singers, their names (which is unique), active since, whether it's a band or not.
- For each album, the singer (band), the year it was recorded, its unique title, its cover photo, and its price must be stored.
- Genres and their hierarchy are also stored. Some genres have emerged from others. For example, a genre can be the parent of 0 or many genres, and another genre can be the child of 0 or 1 genres. Each genre is identified by a unique name.
- A given album may belong to more than one genre, which could be at any level in the hierarchy.
- Songs of each album are to be stored. For each song, the company stores its title and duration, and the album/albums it belongs to, i.e., all songs must be in some album, but can be in many albums.
- Finally, the record company keeps the information about the managers. For each manager, the company keeps the name, address, the total amount of dollars owed. Managers can manage multiple artists and genres. We still store the managers even if they are no longer managing genres/artists.

Draw a conceptual model using Chen's notation for the record label company. Be sure to indicate the various attributes of each entity and relationship set; also specify the key and participation constraints for each relationship set. Be sure to write down any assumptions you make.

Solution

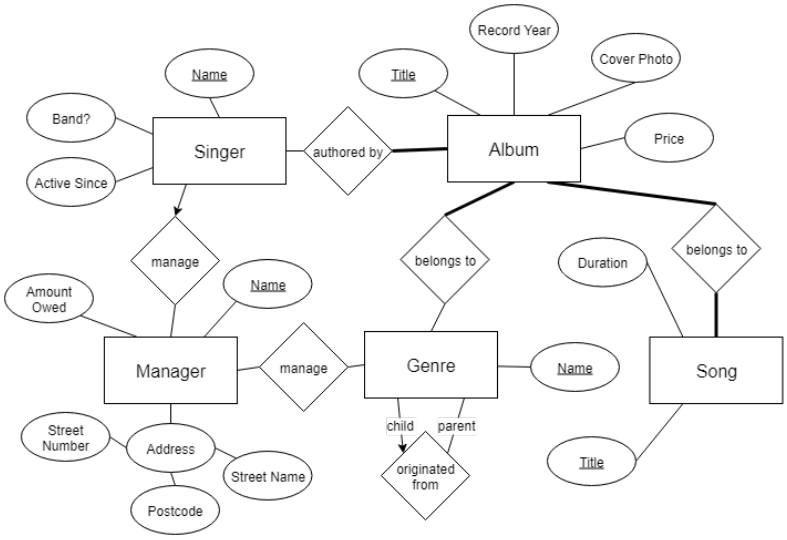


Figure 2: Record Company Conceptual Model

Assumptions:

- Album can be authored by more than one band/singer, e.g., compilation albums.

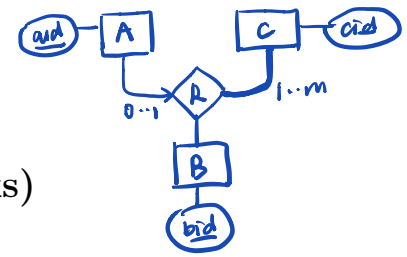
Marking Scheme

Total marks: 15

- Entities: 5 marks
  1 mark for each table
  Deduct 1 mark for each unnecessary entity

- Attributes: 5 marks
  Max of 1 mark for each entity's attributes (summed)
  - 1 mark for each entity that has ALL attributes
  - 0.5 for each entity that has some attributes
- Correct Key Constraints: 2 marks
- Correct Participation Constraints: 2 marks
- Legibility: 1 mark

MARKER NOTE: Ensure that before taking marks off, you check if they have any reasonable assumptions which allows them to change the constraints. E.g. do they make the assumption that Album has only one band/singer?



0.2 ER Modelling II (Exam Q9 - 8 marks)

0.2.1 Option 1

Write the CREATE TABLE statements resulting from the following conceptual model:

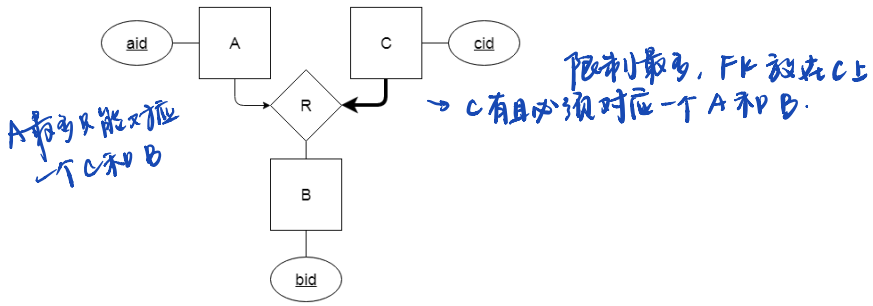


Figure 3: DDL Option 1

Be sure to specify primary and foreign keys. You NEED to specify whether the fields are NULL/NOT NULL. Where the data type is not obvious, feel free to choose an appropriate data type.

Solution

```
CREATE TABLE A (
    aid INT NOT NULL,
    PRIMARY KEY (aid)
);
CREATE TABLE B (
    bid INT NOT NULL,
    PRIMARY KEY (bid)
);
CREATE TABLE C (
    cid INT NOT NULL,
    aid INT NOT NULL UNIQUE,
    bid INT NOT NULL,
    PRIMARY KEY (cid),
    FOREIGN KEY (aid) REFERENCES A (aid),
    FOREIGN KEY (bid) REFERENCES B (bid)
);
```

0.2.2 Option 2

Write the CREATE TABLE statements resulting from the following conceptual model:

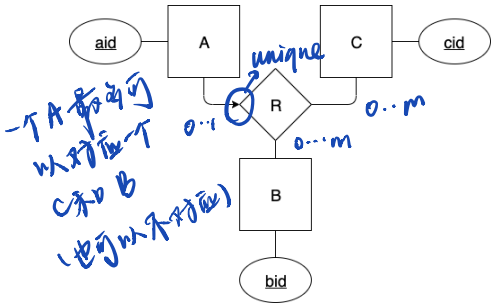


Figure 4: DDL Option 2

Be sure to specify primary and foreign keys. You NEED to specify whether the fields are NULL/NOT NULL. Where the data type is not obvious, feel free to choose an appropriate data type.

Solution

```
CREATE TABLE B (
    bid INT NOT NULL,
    PRIMARY KEY (bid)
);
CREATE TABLE C (
    cid INT NOT NULL,
    PRIMARY KEY (cid)
);
// need extra table rather than 2 optional FKs in A, since that
// would allow one to be null. Also need aid in this relationship so that we
// only allow for ternary relationships (a solution with a FK A to R and R with
// bid + cid only allows for a binary relationship between B+C in R
CREATE TABLE R (
    aid INT NOT NULL,
    bid INT NOT NULL,
    cid INT NOT NULL,
    // a needs to be unique
    PRIMARY KEY (aid),
    FOREIGN KEY (aid) REFERENCES A (aid),
    FOREIGN KEY (bid) REFERENCES B (bid),
    FOREIGN KEY (cid) REFERENCES C (cid)
);
CREATE TABLE A (
    aid INT NOT NULL,
    PRIMARY KEY (aid)
);
```

Note: other options for cases with more key constraints like option 4 won't work here...

0.2.3 Option 3

Write the CREATE TABLE statements resulting from the following conceptual model:

Be sure to specify primary and foreign keys. You NEED to specify whether the fields are NULL/NOT NULL. Where the data type is not obvious, feel free to choose an appropriate data type.

Solution

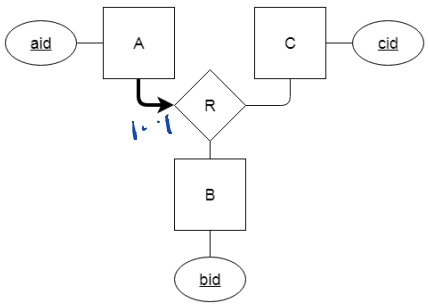


Figure 5: DDL Option 3

```
CREATE TABLE B (
    bid INT NOT NULL,
    PRIMARY KEY (bid)
);
CREATE TABLE C (
    cid INT NOT NULL,
    PRIMARY KEY (cid)
);
CREATE TABLE A (
    aid INT NOT NULL,
    bid INT NOT NULL,
    cid INT NOT NULL,
    PRIMARY KEY (aid),
    FOREIGN KEY (bid) REFERENCES B (bid),
    FOREIGN KEY (cid) REFERENCES C (cid)
);
```

0.2.4 Option 4

Write the CREATE TABLE statements resulting from the following conceptual model:

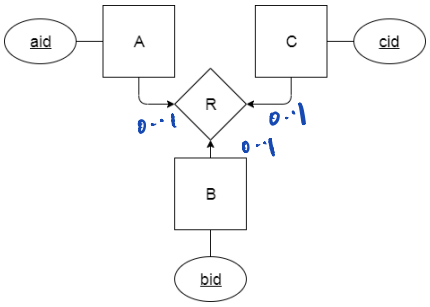


Figure 6: DDL Option 4

Be sure to specify primary and foreign keys. You NEED to specify whether the fields are NULL/NOT NULL. Where the data type is not obvious, feel free to choose an appropriate data type.

Solution

```
CREATE TABLE A (
    aid INT NOT NULL,
    PRIMARY KEY (aid)
);
CREATE TABLE B (
    bid INT NOT NULL,
    PRIMARY KEY (bid)
);
CREATE TABLE C (
    cid INT NOT NULL,
    PRIMARY KEY (cid)
);
CREATE TABLE R (
    aid INT NOT NULL UNIQUE,
    bid INT NOT NULL UNIQUE,
    cid INT NOT NULL UNIQUE,
    PRIMARY KEY (aid, bid, cid),
    FOREIGN KEY (aid) REFERENCES A (aid),
    FOREIGN KEY (bid) REFERENCES B (bid),
    FOREIGN KEY (cid) REFERENCES C (cid)
);
```

Alternative Solution I

```
CREATE TABLE A (
    aid INT NOT NULL,
    rid INT UNIQUE,
    PRIMARY KEY (aid),
    FOREIGN KEY (rid) REFERENCES B (rid),
    FOREIGN KEY (rid) REFERENCES C (rid)
    );
CREATE TABLE B (
    bid INT NOT NULL,
    rid INT UNIQUE,
    PRIMARY KEY (bid),
    -- FOREIGN KEY (rid) REFERENCES A (rid),
    -- FOREIGN KEY (rid) REFERENCES C (rid)
);
CREATE TABLE C (
    cid INT NOT NULL,
    rid INT UNIQUE,
    PRIMARY KEY (cid),
    -- FOREIGN KEY (rid) REFERENCES A (rid),
    -- FOREIGN KEY (rid) REFERENCES B (rid)
);
```

Alternative Solution II [requires constraining ids]

```
CREATE TABLE A (
    aid INT NOT NULL,
    rid INT UNIQUE,
    PRIMARY KEY (aid),
    -- This requires that bid and cid are
    -- constrained...
    FOREIGN KEY (rid) REFERENCES B (bid),
    FOREIGN KEY (rid) REFERENCES C (cid)
    );
CREATE TABLE B (
    bid INT NOT NULL,
    PRIMARY KEY (bid),
);
CREATE TABLE C (
    cid INT NOT NULL,
    PRIMARY KEY (cid),
 );
```

Marking Scheme

Total marks: 8

- Correct implementation of Key Constraints (FK positioning + UNIQUE) : 3 marks
  - No demonstration of dealing with key constraints (even if many are 'incidentally' correct): -2 marks
  - Didn't capture ternary relationship constraint that one set of (a,b,c) participates only once: -1 mark
  - Would have received 0 marks for this and next criteria (ie 0/6), but did have some FK notation present: 1 mark
- Correct implementation of Participation Constraint (NULL/NOT NULL): 3 marks

  - No demonstration of dealing with participation constraints (even if many are 'incidentally' correct): -2 marks
  - Allowed for other binary relations in model (ie no participation constraint that ALL of a,b,c participate in a relationship): -1 mark

- Working table implementation (can store attributes): 1 marks
- Uses correct syntax + PKs for tables: 1 mark

0.3 SQL

Australia has carried out several censuses. The following diagram shows the tables corresponding to preferred means of transportation, residency location, and origins and destinations of the frequent trips.

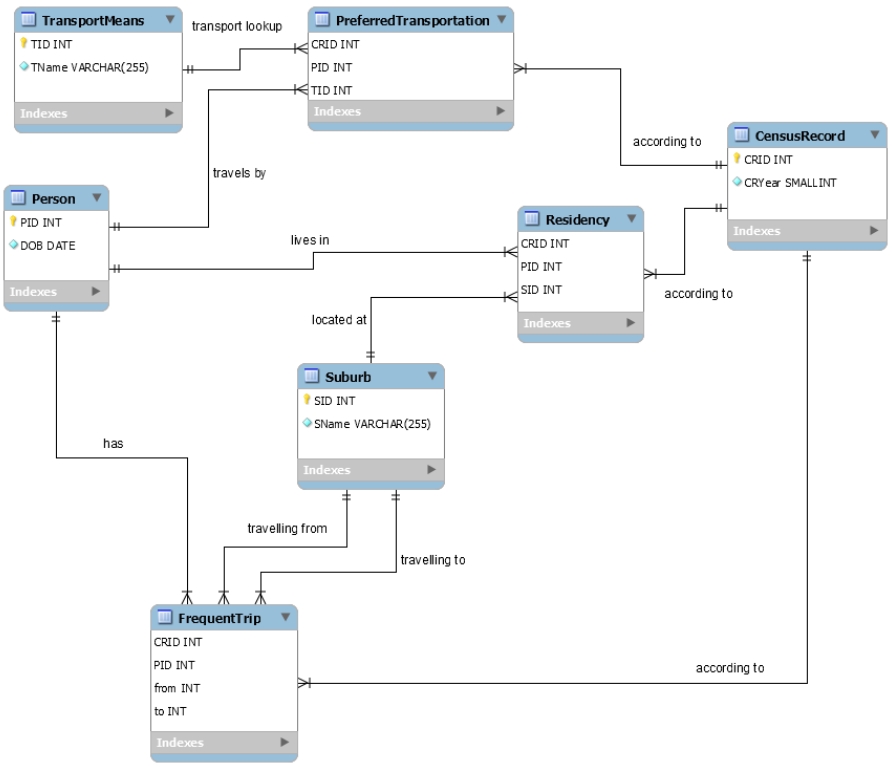


Figure 7: Australia's Censuses Model

Write a single SQL statement to correctly answer each of the following questions. DO NOT USE VIEWS or VARIABLES to answer questions. Query nesting is allowed.

0.4 SQL I (Exam Q10 - 2 marks)

0.4.1 Option 1

How many different people, across all censuses, have said they live in Brunswick?

Solution

```
SELECT  COUNT(DISTINCT R.PID)
FROM    residency AS R
        NATURAL JOIN suburb AS S
WHERE   S.SName = 'Brunswick';
```

0.4.2 Option 2

How many different people, across all censuses, have said they prefer public transport?

Solution

```
SELECT  COUNT(DISTINCT PT.PID)
FROM    preferredtransportation AS PT
        NATURAL JOIN transportmeans AS TM
WHERE   TM.TName = 'Public Transport';
```

0.4.3 Option 3

How many different people, across all censuses, have said they frequently travel to Brunswick?

Solution

```
SELECT  COUNT(DISTINCT FT.PID)
FROM    frequenttrip AS FT
        INNER JOIN suburb AS S ON FT.to = S.SID
WHERE   S.SName = 'Brunswick';
```

Marking Scheme

Total marks: 2

- Correct Answer: 2 marks
  - Missing COUNT: -0.5
  - Incorrect JOIN: -1
  - Missing/Incorrect condition: -0.5
  - Missing DISTINCT: -0.5
  - Other error(s): -0.5 (each)

0.5 SQL II (Exam Q11 - 3 marks)

0.5.1 Option 1

List the name of the suburbs and the number of people who lives in each, according to the 2016 census, ordered from the most to the least populated.

Solution

```
SELECT  S.SName
    ,   COUNT(R.PID)
FROM    suburb AS S
        NATURAL JOIN residency AS R
        NATURAL JOIN censusrecord AS CR
WHERE   CR.CRYear = 2016
GROUP BY
        S.SName
ORDER BY
        COUNT(R.PID) DESC;
```

MARKER NOTE: OK to group by S.SID instead, and thus have multiple rows with suburbs of the same names. Grouping just by name as done above is OK too.

0.5.2 Option 2

List the name of the transport means and the number of people who use each, according to the 2016 census, ordered from the most to the least preferred.

Solution

```
SELECT  TM.TName
    ,   COUNT(PT.PID)
FROM    transportmeans AS TM
        NATURAL JOIN preferredtransportation AS PT
        NATURAL JOIN censusrecord AS CR
WHERE   CR.CRYear = 2016
GROUP BY
        TM.TName
ORDER BY
        COUNT(PT.PID) DESC;
```

Marking Scheme

Total marks: 3

- Correct Answer: 3 marks
  - Missing COUNT: -0.5
  - Incorrect JOIN: -1.5
  - Missing/Incorrect condition: -0.5
  - Missing GROUP BY: -1
  - Missing/Incorrect ORDER BY: -0.5
  - Other error(s): -0.5 (each)

0.6 SQL III (Exam Q12 - 4 marks)

0.6.1 Option 1

It was found that the suburbs that are the worst affected by COVID-19 are Brunswick, Epping, and Port Melbourne. Calculate how many different people, according to the 2016 census, frequently travels from or to those suburbs.

Solution

```
SELECT  COUNT(DISTINCT FT.PID)
FROM    frequenttrip AS FT
        NATURAL JOIN censusrecord AS CR
        INNER JOIN suburb AS S_FROM ON FT.from = S_FROM.SID
        INNER JOIN suburb AS S_TO ON FT.to = S_TO.SID
WHERE   (
            S_FROM.SName IN ('Brunswick', 'Epping', 'Port Melbourne')
            OR
            S_TO.SName IN ('Brunswick', 'Epping', 'Port Melbourne')
        )
AND     CR.CRYear = 2016;
```

0.6.2 Option 2

How many different people, according to the 2016 census, frequently travels from Brunswick to a suburb other than Epping and Port Melbourne?

Solution

```
SELECT  COUNT(DISTINCT FT.PID)
FROM    frequenttrip AS FT
        NATURAL JOIN censusrecord AS CR
        INNER JOIN suburb AS S_FROM ON FT.from = S_FROM.SID
        INNER JOIN suburb AS S_TO ON FT.to = S_TO.SID
WHERE   CR.CRYear = 2016
AND     S_FROM.SName = 'Brunswick'
AND     S_TO.SName NOT IN ('Epping', 'Port Melbourne');
```

0.6.3 Option 3

How many different people who live in Brunswick frequently travel to the CBD, according to the 2016 census?

Solution

```
SELECT  COUNT(DISTINCT R.PID)
FROM    frequenttrip AS FT
        NATURAL JOIN residency AS R
        NATURAL JOIN censusrecord AS CR
        INNER JOIN suburb AS S_LIVE ON R.SID = S_LIVE.SID
        INNER JOIN suburb AS S_TO ON FT.to = S_TO.SID
WHERE   CR.CRYear = 2016
AND     S_LIVE.SName = 'Brunswick'
AND     S_TO.SName = 'CBD';
```

Marking Scheme

Total marks: 4

- Correct Answer: 4 marks
  - Missing COUNT: -0.5
  - Incorrect JOIN: -2
  - Missing/Incorrect condition: -1.5 (-0.5 for each)
  - Other error(s): -0.5 (each)

0.7 SQL IV (Exam Q13 - 6 marks)

0.7.1 Option 1

Of people who prefer public transport, from which suburb do the largest number of different people make frequent trips, according to the 2016 census?

Solution

```
SELECT  S_FROM.SName
    ,   COUNT(FT.PID)
FROM    frequenttrip AS FT
        INNER JOIN suburb AS S_FROM ON FT.from = S_FROM.SID
        NATURAL JOIN preferredtransportation
        NATURAL JOIN censusrecord AS CR
        NATURAL JOIN transportmeans AS TM
WHERE   CR.CRYear = 2016
AND     TM.TName = 'Public Transport'
GROUP BY
        S_FROM.SName
ORDER BY
        COUNT(FT.PID) DESC
LIMIT   1;
```

0.7.2 Option 2

Of people who prefer public transport, in which suburb do the largest number of different people live, according to the 2016 census?

Solution

```
SELECT  S.SName
    ,   COUNT(R.PID)
FROM    residency AS R
        NATURAL JOIN suburb AS S
        NATURAL JOIN preferredtransportation
        NATURAL JOIN censusrecord AS CR
        NATURAL JOIN transportmeans AS TM
WHERE   CR.CRYear = 2016
AND     TM.TName = 'Public Transport'
GROUP BY
        S.SName
ORDER BY
        COUNT(R.PID) DESC
LIMIT   1;
```

Marking Scheme

Total marks: 6

- Correct Answer: 6 marks
  - Missing COUNT: -0.5
  - Incorrect JOIN: -3
  - Missing/Incorrect condition: -1 (-0.5 for each)
  - Missing GROUP BY: -1
  - Missing/Incorrect ORDER BY: -0.5

– Missing LIMIT: -1
– Other error(s): -0.5 (each)

0.8 RA I (Exam Q14 - 2 marks)

Based on the same diagram above (i.e., Australia censuses), write a relational algebra expression for each of the following questions:

0.8.1 Option 1

List the DOB of people whose residency is in Brunswick, according to the 2011 census.

Solution

Any of these:

$\pi_{DOB}(((\sigma_{SName='BSW'}(S) \bowtie R) \bowtie \sigma_{CRYear=2011}(CR)) \bowtie P)$

$\pi_{DOB}(\sigma_{SName='BSW' \wedge CRYear=2011}((S \bowtie R) \bowtie CR) \bowtie P)$

$\pi_{DOB}(\sigma_{SName='BSW' \wedge CRYear=2011}(((S \bowtie R) \bowtie CR) \bowtie P))$

MARKER NOTE: Also consider conditional joins if correct, could include the selection criteria.

0.8.2 Option 2

List the DOB of people whose preferred means of transportation is public transport, according to the 2016 census.

Solution

Any of these:

$\pi_{DOB}(((\sigma_{TName='PT'}(TM) \bowtie PT) \bowtie \sigma_{CRYear=2016}(CR)) \bowtie P)$

$\pi_{DOB}(\sigma_{TName='PT' \wedge CRYear=2016}((TM \bowtie PT) \bowtie CR) \bowtie P)$

$\pi_{DOB}(\sigma_{TName='PT' \wedge CRYear=2016}(((TM \bowtie PT) \bowtie CR) \bowtie P))$

MARKER NOTE: Also consider conditional joins if correct, could include the selection criteria.

Marking Scheme

Total marks: 2

- Correct / Logically Equivalent Answer: 2 marks
  - Correct projection: +0.5
  - Correct selections: +0.5
  - Correct joins: +1

0.9 RA II (Exam Q15 - 3 marks)

Based on the same diagram above (i.e., Australia censuses), write a relational algebra expression for each of the following questions:

0.9.1 Option 1

List the DOB of people who prefer public transport and frequently travel to 'Brunswick', according to the 2016 census.

Solution

$\pi_{DOB}(((((\sigma_{TName='PT'}(TM) \bowtie PT) \bowtie \sigma_{CRYear=2016}(CR)) \bowtie FT) \bowtie_{to=SID} \sigma_{SName='BSW'}(S)) \bowtie P)$

MARKER NOTE: Also consider conditional joins if correct, could include the selection criteria.

0.9.2 Option 2

List the DOB of people who live in 'Brunswick' and prefer public transport, according to the 2016 census.

Solution

$\pi_{DOB}(((((\sigma_{SName='BSW'}(S) \bowtie R) \bowtie \sigma_{CRYear=2016}(CR)) \bowtie PT) \bowtie \sigma_{TName='PT'}(TM)) \bowtie P)$

MARKER NOTE: Also consider conditional joins if correct, could include the selection criteria.

Marking Scheme

Total marks: 3

- Correct / Logically Equivalent Answer: 3 marks
  - Correct projection: +0.5
  - Correct selections: +1
  - Correct joins: +1.5

ist the DOB of people who live in 'Brunswick' and prefer public transport, according to the 2016 census.

0.10 Query Processing - Join (Exam Q16 - 6 marks)

0.10.1 Option 1

Consider two relations called *Customer* and *DocumentHeader*. Imagine that the relation *Customer* has 250 pages and *DocumentHeader* has 1000 pages. Consider the following SQL statement:

```
SELECT  *
FROM    Customer LEFT JOIN DocumentHeader
ON      Customer.CID = DocumentHeader.CID
WHERE   Customer.CName = 'ACME';
```

There are 502 buffer pages available in memory. Both relations are stored as simple heap files. Neither relation has any indexes built on it.

Evaluate block-oriented NLJ, SMJ and HJ using the number of disk I/O's as the cost. Provide the formulae you use to calculate your cost estimates.

Which join would be the optimizer's choice? Consider *Customer* as the outer relation in all alternatives. Assume that sorting can be performed in two passes for both relations. All selections are performed on-the-fly after the join.

Solution

$NPages(C) = 250, NPages(DH) = 1000, B = 502, NBlocks(C) = 1$

$Cost_{BNLJ} = NPages(C) + NBlocks(C) \times NPages(DH)$
$Cost_{BNLJ} = 250 + 1 \times 1000 = 1250$ I/Os

$Cost_{SMJ} = Cost_{SORT(C)} + Cost_{SORT(DH)} + Cost_{MERGE(C,DH)}$
$Cost_{SORT(C)} = 2 \times 2 \times NPages(C)$
$Cost_{SORT(DH)} = 2 \times 2 \times NPages(DH)$
$Cost_{MERGE(C,DH)} = NPages(C) + NPages(DH)$
$Cost_{SMJ} = 2 \times 2 \times 250 + 2 \times 2 \times 1000 + 250 + 1000 = 6250$ I/Os

$Cost_{HJ} = 3 \times NPages(C) + 3 \times NPages(DH)$
$Cost_{HJ} = 3 \times 250 + 3 \times 1000 = 3750$ I/Os

Optimizer's choice would be BNLJ.

0.10.2 Option 2

Consider two relations called *DocumentHeader* and *DocumentDetail*. Imagine that the relation *DocumentHeader* has 1000 pages and *DocumentDetail* has 10000 pages. Consider the following SQL statement:

```
SELECT  *
FROM    DocumentHeader RIGHT JOIN DocumentDetail
ON      DocumentHeader.DID = DocumentDetail.DID
WHERE   DocumentHeader.DType = 'Cheque';
```

There are 502 buffer pages available in memory. Both relations are stored as simple heap files. Neither relation has any indexes built on it.

Evaluate block-oriented NLJ, SMJ and HJ using the number of disk I/O's as the cost. Provide the formulae you use to calculate your cost estimates.

Which join would be the optimizer's choice? Consider *DocumentHeader* as the outer relation in all alternatives. Assume that sorting can be performed in three passes for both relations. All selections are performed on-the-fly after the join.

Solution

$NPages(DH) = 1000, NPages(DD) = 10000, B = 502, NBlocks(DH) = 2$

$Cost_{BNLJ} = NPages(DH) + NBlocks(DH) \times NPages(DD)$
$Cost_{BNLJ} = 1000 + 2 \times 10000 = 21000$ I/Os

$Cost_{SMJ} = Cost_{SORT(DH)} + Cost_{SORT(DD)} + Cost_{MERGE(DH,DD)}$
$Cost_{SORT(DH)} = 2 \times 3 \times NPages(DH)$
$Cost_{SORT(DD)} = 2 \times 3 \times NPages(DD)$
$Cost_{MERGE(DH,DD)} = NPages(DH) + NPages(DD)$
$Cost_{SMJ} = 2 \times 3 \times 1000 + 2 \times 3 \times 10000 + 1000 + 10000 = 77000$ I/Os

$Cost_{HJ} = 3 \times NPages(DH) + 3 \times NPages(DD)$
$Cost_{HJ} = 3 \times 1000 + 3 \times 10000 = 33000$ I/Os

Optimizer's choice would be BNLJ.

Marking Scheme

Total marks: 6

- Correct calculation of BNLJ: 2 marks
  - Some good work: 0.5 marks
- Correct calculation of SMJ: 2 marks
  - Some good work: 0.5 marks
- Correct calculation of HJ: 1 mark
- Correct lowest cost [NO consequential marks]: 1 mark

0.11 Query Processing - Extra index change cost (Exam Q17 - 2 marks)

Would the cost of Sort-Merge Join have changed if we would have had an unclustered B+ tree index created on the attribute in the WHERE condition? If yes, explain how (no need to recalculate). If no, explain why.

Solution

There are three accepted answers:

1. No effect. Unclustered indexes = data file not sorted so not going to affect SMJ.
2. No effect. Index is not on the join attribute so even if sorted not going to affect SMJ.
3. Yes. The data could be filtered before the join takes place, reducing the number of pages entering the SMJ.

Marking Scheme

Total marks: 2

- Correct Answer & Reason: 2 marks
  - Gave incorrect reason: 0/2 marks

0.12 Query Processing - Indexes Question

0.12.1 Option 1

Consider a relation called Loans that stores information about loans given by a bank. Imagine that the relation Loans consists of 800 pages and each page stores 50 tuples. Imagine that the catalog manager provides statistics about the distribution of some attributes of this relation as the following:

```
Type: {'A': 10%, 'B': 30%, 'C': 45%, 'D': 15%}
ApprovalYear: {2018: 10%, 2019: 60%, 2020: 30%}
```

Suppose that the following SQL query is executed frequently using the given relation:

```
SELECT  Type
FROM    Loans
WHERE   Type IN ('B', 'C')
AND     ApprovalYear > 2019;
```

0.12.2 Option 2

Consider a relation called Loans that stores information about loans given by a bank. Imagine that the relation Loans consists of 500 pages and each page stores 100 tuples. Imagine that the catalog manager provides statistics about the distribution of some attributes of this relation as the following:

```
Type: {'A': 10%, 'B': 25%, 'C': 45%, 'D': 15%, 'E': 5%}
ApprovalYear: {2017: 10%, 2018: 40%, 2019: 40%, 2020: 10%}
```

Suppose that the following SQL query is executed frequently using the given relation:

```
SELECT  Type
FROM    Loans
WHERE   Type IN ('B', 'C')
AND     ApprovalYear > 2018;
```

0.13 Query Processing - Indexes I (Exam Q18 - 2 marks)

Compute the estimated result size (in number of tuples) for the query, and the reduction factor of each predicate.

0.13.1 Solution Option 1

```
RF(Type IN ('B', 'C')) = 0.75
RF(ApprovalYear > 2019) = 0.3
Result size = 800 * 50 * 0.75 * 0.3 = 9,000 tuples
```

0.13.2 Solution Option 2

```
RF(Type IN ('B', 'C')) = 0.7
RF(ApprovalYear > 2018) = 0.5
Result size = 500 * 100 * 0.7 * 0.5 = 17,500 tuples
```

0.13.3 Marking Scheme

Total marks: 2

- Correct Results: 2 marks
  - 1 error (forgive consequential errors): 1/2 marks

0.14 Query Processing - Indexes II (Exam 19 - 2 marks)

0.14.1 Option 1

Compute the estimated cost of plan alternatives and state which plan is the best assuming that an **unclustered** B+ tree index on *(ApprovalYear)* is (the only index) available. Suppose there are 100 index pages.

Solution Option 1

```
Cost = (100 + 800 * 50) * 0.3 = 12,030 I/Os
HeapScan = 800 I/Os
Best option is Heap Scan
```

Solution Option 2

```
Cost = (100 + 500 * 100) * 0.5 = 25,050 I/Os
HeapScan = 500 I/Os
Best option is Heap Scan
```

0.14.2 Option 2

Compute the estimated cost of plan alternatives and state which plan is the best assuming that a **clustered** B+ tree index on *(ApprovalYear)* is (the only index) available. Suppose there are 100 index pages.

Solution Option 1

```
Cost = (100 + 800) * 0.3 = 270 I/Os
HeapScan = 800 I/Os
Best option is Index Scan
```

Solution Option 2

```
Cost = (100 + 500) * 0.5 = 300 I/Os
HeapScan = 500 I/Os
Best option is Index Scan
```

0.14.3 Marking Scheme

Total marks: 2

- Correct Index Scan Cost: 1 mark
- Identifies HS as option + states HS cost + chooses best: 1 mark

0.15 Query Processing - Indexes III (Exam Q20 - 2 marks)

What would happen if our query changed and became:

```
SELECT  *
FROM    Loans
WHERE   2018 < ApprovalYear
AND     ApprovalYear < 2020
AND     Type IN ('B', 'C');
```

Will this change impact the result size? If yes, calculate the new result size. If no, explain why.

0.15.1 Solution Option 1

Yes, it will change the result size.

```
[Unchanged] RF(Type IN ('B', 'C')) = 0.75
[New] RF(2020 > ApprovalYear > 2018) = 0.6
Result size = 800 * 50 * 0.75 * 0.6 = 18,000 tuples
```

0.15.2 Solution for Question option II

Yes, it will change the result size.

```
[Unchanged] RF(Type IN ('B', 'C')) = 0.7
[New] RF(2020 > ApprovalYear > 2018) = 0.4
Result size = 500 * 100 * 0.7 * 0.4 = 14,000 tuples
```

0.15.3 Marking Scheme

Total marks: 2

- "Yes it will change the result size": 1 marks
- Correct new result size calculated: 1 marks

0.16 Query Optimization Question

0.16.1 Option 1

Consider three relations called *Account*, *AccountDetail*, and *TransactionType*. Imagine that the relation *Account* has 500 pages, *AccountDetail* has 5000 pages, and *TransactionType* has 100 pages. Each page stores 100 tuples. The *AccountType* attribute has values in the set {*'Savings', 'CreditCard'*}. There is a clustered B+ tree index on *AccountType*, which is 100 pages in size. Consider the following query:

```
SELECT  *
FROM    Account AS A, AccountDetail AS AD, TransactionType AS TT
WHERE   A.Account_ID = AD.Account_ID AND AD.TranType_ID = TT.TranType_ID
AND     A.AccountType = 'Savings';
```

0.16.2 Option 2

Consider three relations called *Account*, *AccountDetail*, and *TransactionType*. Imagine that the relation *Account* has 250 pages, *AccountDetail* has 5000 pages, and *TransactionType* has 50 pages. Each page stores 100 tuples. The *AccountType* attribute has values in the set {*'Savings', 'CreditCard'*}. There is a clustered B+ tree index on *AccountType*, which is 50 pages in size. Consider the following query:

```
SELECT  *
FROM    Account AS A, AccountDetail AS AD, TransactionType AS TT
WHERE   A.Account_ID = AD.Account_ID AND AD.TranType_ID = TT.TranType_ID
AND     A.AccountType = 'Savings';
```

0.17 Query Optimisation I (Exam Q21 - 4 marks)

Compute the cost of the plan shown below. NLJ is a Page-oriented Nested Loops Join. Assume that *Account_ID* is the candidate key of *Account*, *TranType_ID* is the candidate key of *TransactionType*, and 100 tuples of a resulting join between *Account* and *AccountDetail* can fit on one page.

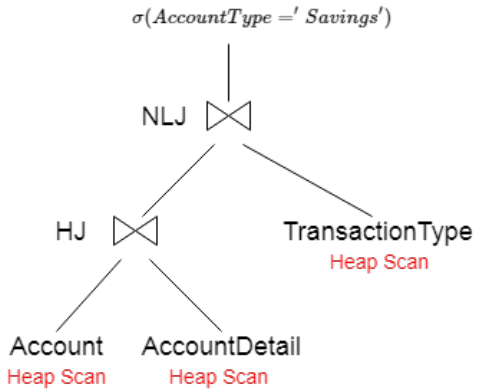


Figure 8: Query Optimisation I

0.17.1 Solution Option 1

```
Cost HJ = 3 * 500 + 3 * 5,000 = 16,500 I/O
Size HJ result = 500 * 100 * 5,000 * 100 * 1/(500 * 100) = 500,000 tuples
               = 5,000 pages
Cost NLJ = 5,000 * 100 = 500,000 I/O
Cost Selection = 0 I/O (pipelined/on the fly)
Total cost = 516,500 I/O
```

0.17.2 Solution Option 2

```
Cost HJ = 3 * 250 + 3 * 5,000 = 15,750 I/O
Size HJ result = 250 * 100 * 5,000 * 100 * 1/(250 *100) = 500,000 tuples
               = 5,000 pages
Cost NLJ = 5,000 * 50 = 250,000 I/O
Cost Selection = 0 I/O (pipelined/on the fly)
Total cost = 265,750 I/O
```

0.17.3 Marking Scheme

MARKER NOTE: Apply consequential marks to all sections aside from 'Total Correct' if correct formulae are given and otherwise correct substitutions are made. E.g. A student incorrectly calculates 'Result Size', so they receive 0 for that component. Their NLJ calculation includes formulae and is correct other than the use of this Result Size, so they still receive the NLJ mark. They do NOT receive the 'Total Correct' mark, so they'd finish with a mark of 2/4.

Total marks: 4

- HJ cost correct: 1 mark
- Result size correct: 1 mark
- NLJ cost correct [consequential]: 1 mark
- Total correct (+ no selection cost) [NO consequential]: 1 mark

0.18 Query Optimisation II (Exam Q22 - 8 marks)

Compute the cost of the plan shown below. Assume that SMJ can be done in 2 passes. Assume that *Account_ID* is the candidate key of *Account*, *TranType_ID* is the candidate key of *TransactionType*, and 100 tuples of a resulting join between *Account* and *AccountDetail* can fit on one page.

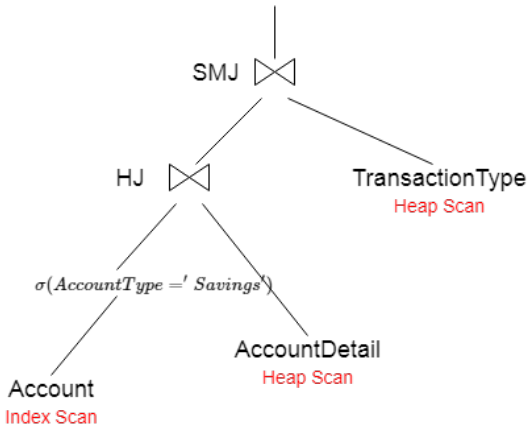


Figure 9: Query Optimisation II

0.18.1 Solution Option 1

```
Cost Selection = (500 + 100) * 1/2 = 300 I/O
Size Selection result = 500 * 1/2 = 250 pages
Cost HJ = 2 * 250 + 3 * 5,000 = 15,500 I/O
Size HJ result = 250 * 100 * 5,000 * 100 * 1/(500 * 100) = 250,000 tuples
               = 2,500 pages
Cost SMJ = 2 * 2 * 2,500 + 2 * 2 * 100 + 100 = 10,500 I/O
Total cost = 26,300 I/O
```

0.18.2 Solution Option 2

```
Cost Selection = (250 + 50) * 1/2 = 150 I/O
Size Selection result = 250 * 1/2 = 125 pages
Cost HJ = 2 * 125 + 3 * 5,000 = 15,250 I/O
Size HJ result = 125 * 100 * 5,000 * 100 * 1/(250 * 100) = 250,000 tuples
               = 2,500 pages
Cost SMJ = 2 * 2 * 2,500 + 2 * 2 * 50 + 50 = 10,250 I/O
Total cost = 25,650 I/O
```

0.18.3 Marking Scheme

MARKER NOTE: Many people are going to miss the correct selection result size and thus have many consequential errors. Thus, if they provided correct formulae and substituted in correctly, they receive full marks for later sections, aside from the TOTAL (see Q21 markscheme for example).

Total marks: 8

- Reduction Factor: 1 mark
- Selection cost: 1 mark
- Selection result size: 1 mark
- HJ cost: 1 mark

- HJ result size: 2 marks
- SMJ cost: 1 mark
- Total [NO consequential]: 1 mark

0.19 Query Optimisation III (Exam Q23 - 3 marks)

Would the cost of the plan presented in the previous question change if the selection on *AccountType* happened between HJ and SMJ, or after SMJ? Explain (intuitively) which out of the three possibilities would be the cheapest: a) the original plan presented in the previous question, b) the plan with selection happening between HJ and SMJ, or c) the plan with selection happening after SMJ as the last operation in the plan. You do not need to calculate the cost of all three alternatives, an explanation in 2-3 sentences is sufficient.

0.19.1 Solution

The above presented plan is intuitively the most efficient.
It filters out tuples which don't meet the selection criteria earlier on, and thus reduces the cost of BOTH HJ and SMJ.

Note if the student goes into more details it is OK, eg:
Because it's a CLUSTERED B+ tree, it's cheap to perform this selection. The size of the index times the RF is always less than the amount we save performing the access itself (without even considering the benefits of the joins) since the RF is 0.5 and the index is smaller than the *Account* entity, thus Heapscan is more expensive than Index Access. (if it was e.g. an unclustered B+ tree, might be cheaper to just not use it and do on the fly selection later!)

0.19.2 Marking Scheme

Total marks: 3

- Plan presented above cheapest: 1 mark
- Selecting earlier reduces size for joins later: 2 marks

0.20 Normalisation

0.20.1 Option 1

You work at PricewaterhouseCoopers and you are auditing the *AccountingDocs* relation of a client. The table below contains the list of accounting documents:

TranNum	Type	Customer	Bank	BSB	AccNum	Branch Address	Year	Month	Status
15013	3	ACME	CBA	123-456	1456784	4 Roberts, 3055	2019	12	approved, printed
15014	3	IBM	NAB	455-788	789465664	10 Grattan St, 3000	2019	12	approved, printed
24604	2	NIKKON	CBA	123-789	42457-Q	55 Brunswick St, 3057	2019	12	approved
24605	2	ACME	CBA	123-456	1456784	4 Roberts, 3055	2019	12	rejected
15015	3	MS	ANZ	788-777	468646	88 Spencer St, 3000	2019	12	rejected
15014	3	IBM	NAB	455-788	789465664	10 Grattan St, 3000	2020	06	rejected
24606	2	DBX	ANZ	788-566	456866	10 Collins St, 3000	2020	05	approved

Figure 10: NF question option 1

The set $\{Year,\ TranNum\}$ is a candidate key for this relation. The following functional dependencies hold for this relation:

$Year, TranNum \rightarrow Type, Customer, BSB, AccNum, Status$
$BSB, AccNum \rightarrow Customer, Bank, BranchAddress$
$BSB \rightarrow Bank, BranchAddress$

0.20.2 Option 2

You work in the IT department at CBA. The relation *LoanSchedules* below contains the amortization schedules of the loans approved by the bank:

DueDate	Status	Principal	Interest	Balance	LoanNum	ApprovalDate	Score	Client	CID
31/01/2019	paid	100	12	1007	4885	01/01/2019	B1	James B	7846564
28/02/2019	partial, past due	101	11	950	4885	01/01/2019	B1	James B	7846564
31/01/2019	paid	50	40	4561	4886	02/01/2019	A2	Ben R	8758155
28/02/2019	paid	52	38	4509	4886	02/01/2019	A2	Ben R	8758155
31/12/2018	paid	1550	450	10020	4710	01/12/2018	B2	James B	7846564
31/01/2019	partial, past due	1565	435	10000	4710	01/12/2018	B2	James B	7846564
28/02/2019	past due	1580	420	10000	4710	01/12/2018	B2	James B	7846564

Figure 11: NF question option 2

The set {*LoanNum, DueDate*} is a candidate key for this relation. The following functional dependencies hold for this relation:

$LoanNum, DueDate \rightarrow Status, Principal, Interest, Balance$
$LoanNum \rightarrow CID, ApprovalDate, Score$
$CID \rightarrow Client$

0.21 Normal Form I (Exam Q24 - 5 marks)

Could any anomalies arise in the table above? If yes, discuss which anomalies and problems may appear in this table. Provide examples.

0.21.1 Solution

All 3 types of anomalies (insertion, deletion, update) can occur in either case.
Students need to name + provide examples for each type

0.21.2 Marking Scheme

Total marks: 5

- States anomalies are present: 2 marks
  - States some anomaly types (not all) that occur: 1/2 marks
  - States all anomaly types and asserts they can occur: 2/2 marks
- Provides examples of anomaly instances : 3 marks
  - 1 mark for valid example per anomaly type

0.22 Normal Form II (Exam Q25 - 10 marks)

Normalize the table above to 3rd Normal Form (3NF). For each step, explicitly identify which normal form is violated and briefly explain why. Write the normalised tables in textual format, as in:

TableName (PrimaryKey, Column, ForeignKey(FK))
AnotherTable (PrimaryKey, Column, AnotherColumn)

0.22.1 Option 1

Solution

The data given is NOT in 1st NF as there are multiple values in the Status column.

Normalising this gives:

AccountingDocs (Year, TranNum, Type, Customer, BSB, AccNum, Bank, BranchAddress, Month)
AccountingDocsStatus(Year(FK), TranNum(FK), Status)

The data given is already in 2nd NF as there are no partial dependencies.

The data given is NOT in 3rd NF as there are transitive dependencies. Removing the first transitive dependency and bringing along its dependents gives:

AccountingDocs (Year, TranNum, Type, BSB(FK), AccNum(FK), Month)
AccountingDocsStatus(Year(FK), TranNum(FK), Status)
CustAcct (BSB, AccNum, Customer, Bank, BranchAddress)

MARKER NOTE: FKs might be placed on different tables and remain technically correct.

The students need to state which normal form they are in at this point. If the student dealt with the BSB transitive dependency first, then they are still in 2nd NF and have one more transitive dependency to normalise. If, however, as done here, they started with the BSB,AccNum dependency, then they are no longer in 2nd NF and the dependency remaining is a partial functional dependency.
Either way, removing the last dependency yields:

AccountingDocs (Year, TranNum, Type, BSB(FK), AccNum(FK), Month)
AccountingDocsStatus(Year(FK), TranNum(FK), Status)
CustAcct (BSB(FK), AccNum, Customer)
Bank (BSB, Bank, BranchAddress)

Which is fully normalised/in 3rd normal form as there are no partial/transitive dependencies

0.22.2 Option 2

Solution

The data given is NOT in 1st NF as there are multiple values in the Status column.

Normalising this gives:

LoanSchedules (LoanNum, DueDate, Principal, Interest, Balance, ApprovalDate, Score, Client, CID)
LoanStatus (LoanNum(FK), DueDate(FK), Status)

MARKER NOTE: FKs might be placed on different tables and remain technically correct.

We are now in 1st NF since the data is tabular. We are NOT in the 2nd NF since there is a partial dependency. Removing this and extracting the dependents gives:

LoanSchedules (LoanNum(FK), DueDate, Principal, Interest, Balance)
LoanStatus (LoanNum(FK), DueDate(FK), Status)
Loans(LoanNum, ApprovalDate, Score, CID, Client)

We are in 2nd NF since there are no partial dependencies. We are NOT in 3rd NF since there is a transitive dependency remaining. Removing this and extracting dependents gives:

LoanSchedules (LoanNum, DueDate, Principal, Interest, Balance)
LoanStatus (LoanNum(FK), DueDate(FK), Status)
Loans(LoanNum, ApprovalDate, Score, CID(FK))
Clients(CID, Client)

Which is fully normalised as there are no partial/transitive dependencies

0.22.3 Marking Scheme

Total marks: 10

- Correctly identifies data's NF at each stage: 2 marks
  - 0.5 marks for each identification of NF: at beginning, and after each normalisation step = total of 4 identifications [give consequential marks].
- Correctly converts to 1NF: 2 marks
  - Incorrect final answer, but some 'good work': 1 marks
- Correctly normalises the 1st functional dependency [give consequential marks]: 2 marks
  - Incorrect final answer, but some 'good work': 1 marks

- Correctly normalises the 2nd functional dependency [give consequential marks]: 2 marks
  - Incorrect final answer, but some 'good work': 1 marks
- Correct final set of tables [NO consequential marks]: 2 marks

0.23 Normalisation III (Exam Q26 - 2 marks)

Normalisation prevents the anomalies as discussed above, however, some databases are deliberately non-normalised. Explain in 2-3 sentences why/when we might choose to not normalise a database.

0.23.1 Solution

- Primary benefit of denormalised database is speed: fewer joins = faster access. This is beneficial when accessing data frequently.
- Less need to normalise if data is historical and not altered once entered (eg a Data Warehouse) since not having to worry about eg updating data in many places since we're not updating data.
- Less need to normalise if Insertion operations on the DB are automated and data is standardised as part of this process (as is the case in DataWarehouses)

0.23.2 Marking Scheme

Total marks: 2

- Correctly states primary benefit of speed: 1 mark
- Gives one other decent reason/situation: 1 mark

0.24 Transactions (Exam Q30 - 6 marks)

Alice and Bob are two Unimelb students who are both interested in buying a new Nintendo Switch (to deal with isolation boredom). They've both wound up on the JB Hifi website, where the page lists that there is only 1 left in stock. Both Alice and Bob are trying to reserve the switch so they can go pick it up in store.

The database table might look something like this:

StockReservations(StockID, StoreID(FK), ItemTypeID(FK), ReservationName)

The timeline of their interactions with the site is as follows:

Time t	Bob	Alice
t =0	Clicks the JB Hifi link from google [reads available unreserved stock]	
t=1		Clicks the JB Hifi link from google [reads available unreserved stock]
t=2		Clicks 'Reserve' [writes new ReservationName]
t=3	Clicks 'Reserve' [writes new ReservationName]	

Figure 12: Transactions Timeline

Answer the following questions:

1. What is the name for this sort of common problem?
2. Who ends up with their name on the switch reservation after this interaction?
3. Give two possible ways to implement concurrency control to prevent this issue, and for each solution, indicate which of Bob and Alice will wind up with the Nintendo Switch!

0.24.1 Solution

1. This is the Lost Update Problem
2. Bob has the switch reserved
3. A few possible options:
   - Locking: Bob or Alice could get the switch depending on implementation... eg if binary lock = Bob wins, but if has separate R/W locks can get more complicated (eg potential deadlock!)
   - Timestamping: Alice gets the switch
   - Optimism: Alice gets the switch

0.24.2 Marking Scheme

Total marks: 6

- Correctly identifies 'Lost Update Problem' : 1 mark

- Correctly identifies Bob gets switch: 1 mark
- Gives 2 correct Concurrency Control methods: 2 marks
- Correctly identifies who will have reserved the switch for each concurrency control method identified: 2 marks

0.25 Data Warehousing

0.25.1 Option 1

A company wants to create a data warehouse that enables analysing the performance of its resellers, which is measured by the sales amount and number of products sold. The company also wants to know the net income (the difference between the sales amount and the product list price.) The manager wants to be able to filter the report by product (name, description, color, list price), product subcategory (name), and product category (name). Since the company has resellers across the country, it is important to filter by state and city as well. Of course, having a filter by the reseller (name, address) is a must. The report can also be filtered by year, semester, or quarter based on either the order date, due date, or ship date. The employee (SSN, name, DOB) who is in contact with the reseller is also an important filter.

0.25.2 Option 2

You work for AirQuality Australia and you have to create a data warehouse to keep track of the pollutants present in the air. The main air pollutants and their units of measurement are: (a) particulate matter PM10 and PM2.5 ($\mu g/m^3$), (b) ground-level ozone O3 (ppm), (c) nitrogen dioxide NO2 (ppb), (d) carbon monoxide CO (ppm), and (e) sulphur dioxide SO2 (ppb). Sources of these pollutants are motor vehicle exhaust, industry, wildfires, etc. Sensors of different types have been set up across the country to measure the levels of these pollutants. Each sensor location is registered as a latitude and longitude pair, which in turn, is mapped to its corresponding suburb. There are even specialized satellites that report the levels of contamination. In the case of satellites, latitude and longitude are left empty. Public health institutions are interested in knowing the levels of contamination aggregated by sensor location, type of sensor, and pollutant. Due to the massive wildfires in Australia that occurred recently, authorities want to have daily and weekly reports at their disposal.

0.26 Data Warehousing I (Exam Q31 - 5 marks)

Use **Kimball's four-step dimensional design process** for designing a dimensional model to support the design of this data warehouse. Explain what you achieved in each step of Kimball's model. In particular make sure to discuss the name of the business process, identify and explain dimensions, declare the grain, and identify and explain facts.

0.26.1 Option 1

Solution

1. Choose a business process: Sale
2. Choose facts: Sale amount, Number of products sold, Net income.
3. Choose the granularity: Group by individual product, group by city, group by individual resellers, group by quarter for all time details, group by individual employee.
4. Choose dimensions: Product, Reseller, Employee, Time

0.26.2 Option 2

Solution

1. Choose a business process: Air Pollution Measurement.
2. Choose facts: Pollutant Measurement.
3. Choose the granularity: Group by individual pollutant type, group by sensor location, group by type of sensor, group by day.
4. Choose dimensions: Time, Pollutant, SensorType, SensorLocation

0.26.3 Marking Scheme

Total marks: 5

- Correct business process: 1 mark
- Correct facts: 1 mark
  - extra facts added: -0.5 marks
  - some facts missing: -0.5 marks
- Correct granularities [cannot go negative]: 2 marks
  - Incorrect granularity chosen: -0.5 per incorrect grain
  - Did not specify granularity for important attribute: -1 per missing
- Correct dimensions [cannot go negative]: 1 marks
  - Missed a dimension: -0.5 marks per missed dimension
  - Added extra dimension: -0.5 marks per added dimension

0.27 Data Warehousing II (Exam Q32 - 5 marks)

Draw a *star schema* to support the design of this data warehouse, showing the attributes in each table. Use PK to denote primary key, PFK to denote primary foreign key, and FK to denote foreign key. You do not need to specify data types nor whether the fields are NULL/NOT NULL.

0.27.1 Option 1

Solution

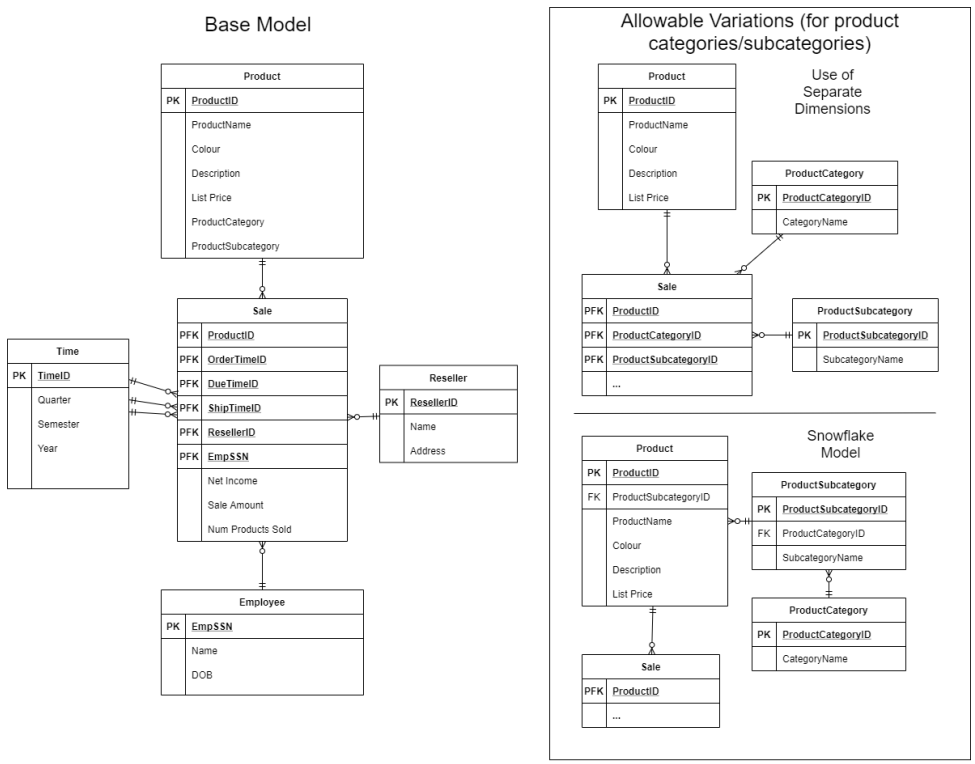


Figure 13: Solution for Question Option 1

0.27.2 Option 2

Solution

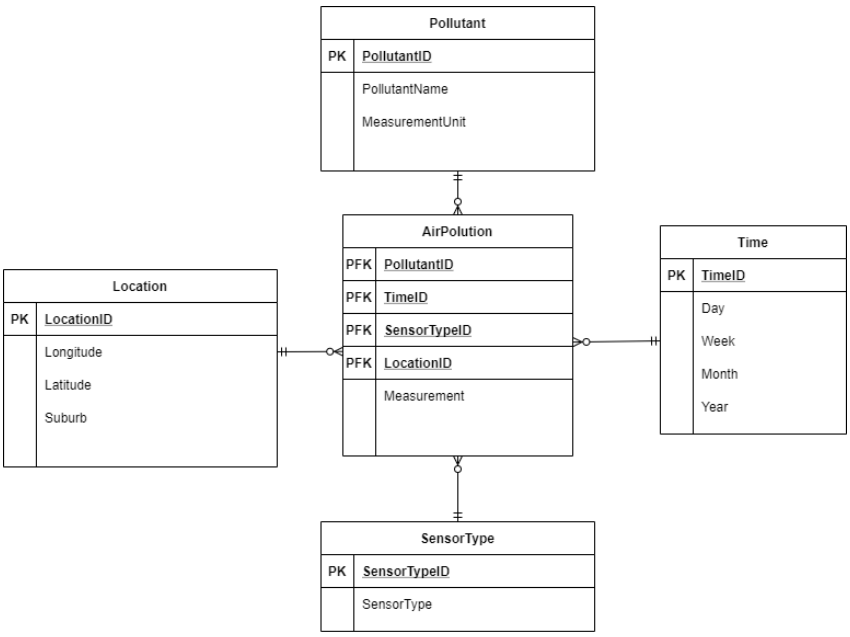


Figure 14: Solution for Question Option 2

0.27.3 Marking Scheme

MARKER NOTE: *all parts of this question are consequential* to the previous question where the model was designed (Q31), *aside* from a final mark for 'overall correctness' which is not consequential.

Total marks: 5

- Correct translation of fact table + PFKs: 1 mark
- Correct translation of dimension tables + PKs + attributes: 2 marks
  - Single mistake (eg missing attribute): -0.5 marks per mistake
  - Consistent mistake across all tables (eg no PKs): -1 mark per mistake
- Correct use of key/participation constraints + notation: 1 mark
- Overall correct solution [NO consequential]: 1 mark

0.28 Distributed Databases (Exam Q33 - 2 marks)

0.28.1 Option 1

Why is "Data Consistency" VERY POOR in a distributed database with "Integrated Partitions"?

DO NOT USE the explanation in the table provided in the lecture slides.

Solution

By definition, all data is not present in every node, so the node is in "constant" inconsistent state. When a user in location A queries info stored in location B, chances are that the presented data is out-of-date based on the assumption that availability is prime (which it usually is).

MARKER NOTE: If the student mentions something about the data being in different locations but without mentioning that the data is out-of-date: 1 mark.

0.28.2 Option 2

What type of distributed database do you think a bank might use, and why? Explain in a few sentences.

Solution

Synchronised replication should be preferred as data consistency is required. A transaction in location A should be reflected as soon as possible in the rest of locations to avoid inconsistencies.

0.28.3 Marking Scheme

Total marks: 2

- Correct answer : 2 marks
  - Answer has some merit: 1 mark

0.29 NoSQL I (Exam Q34 - 2 marks)

0.29.1 Option 1

Give an example of Eventual Consistency (BASE) in NoSQL databases and explain in 2-3 sentences.

Solution

A Facebook user in location B does not see a post from a user in location A. However, after some time the post list is "eventually" refreshed (data is replicated across all locations asynchronously).

Student needs to mention that a) initially there is allowed to be an inconsistency and b) eventually this inconsistency is resolved.

0.29.2 Option 2

Explain in 2-3 sentences how the CAP theorem is related with BASE in NoSQL databases.

Solution

When NoSQL databases are distributed, they tend to favour Availability over Consistency (since can only have 2/3 according to CAP theorem)

BASE is an acronym that captures these tendencies of NoSQL DBs, stating that they are 'Basically Available' (ie they place value on Availability), and Eventually consistent + Soft State covers that over time without input, the inconsistencies across the distributed database will resolve themselves.

0.29.3 Marking Scheme

Total marks: 2

- Correct answer : 2 marks
  - Answer has some merit: 1 mark

0.30 NoSQL II (Exam Q35 - 2 marks)

0.30.1 Option 1

Give an example of how NoSQL databases help solve the Object-Relational impedance mismatch problem.

Solution

In document-based NoSQL databases, JSON documents can mimic the exactly same structure as an object from the Object-Oriented world. A JSON document enables multi-level hierarchy in its information. For example, a list attribute in a JSON document contain items which in turn can be lists. Such a structure is not possible in relational databases.

0.30.2 Option 2

It seems like Schema-on-Read in NoSQL databases is awesome. However, I bet there should be disadvantages with it. Give one disadvantage and explain why in 2-3 sentences.

Solution

Here are some examples:

- Overhead when reading which can be significant as information can have many different formats.
- Difficult to perform analysis on data since we likely have no way of knowing ahead of time which data are relevant and which aren't, need to traverse most of it.

- Likely more space-intensive to store data, since the data INCLUDES the schema information also (and thus we store it many times over)

0.30.3 Marking Scheme

Total marks: 2

- Correct answer : 2 marks
  - Answer has some merit: 1 mark