COMP2400/6240 - Relational Databases

Assignment 2 (Database Theory)

Assignment 2 on Database Theory (Solutions)

Due date: 23:59, 12 October 2020

Instructions:

- **This assignment must be done individually (no group work)**.
- This assignment will count for 15% of the final grade. Marks are assigned for the process of finding a solution, not only for the result. Hence, include all essential ideas and steps that are necessary to derive a solution.
- You must submit a single PDF file named as "u1234567.pdf" (replace u1234567 with your UID). Make sure you only upload a PDF file, not a Word or text file.
- You should try your best to type the solutions. The scanned images of handwritten texts and equations can be unreadable for marking. As for the EER diagram, you are highly recommended to export a JPEG file from TerraER and include it in the PDF file.
- Late submission is not granted under any circumstance. You will be marked on whatever you have submitted at the time of the deadline. Please take careful note of deadlines and adhere to them. Of course, if you find yourself in a situation beyond your control that you believe significantly affects an assessment, you should follow the ANU's special consideration process (http://www.anu.edu.au/students/program-administration/assessments-exams/special-assessment-consideration).
- **Plagiarism will attract academic penalties in accordance with the ANU guidelines. A student in this course is expected to be able to explain and defend any submitted assessment item. The course convener can conduct or initiate an additional interview about any submitted assessment item for any student. If there is a significant discrepancy between the two forms oexpressionsf assessment, it will be automatically treated as a case of suspected academic misconduct.**

Question 1 **4 Marks**

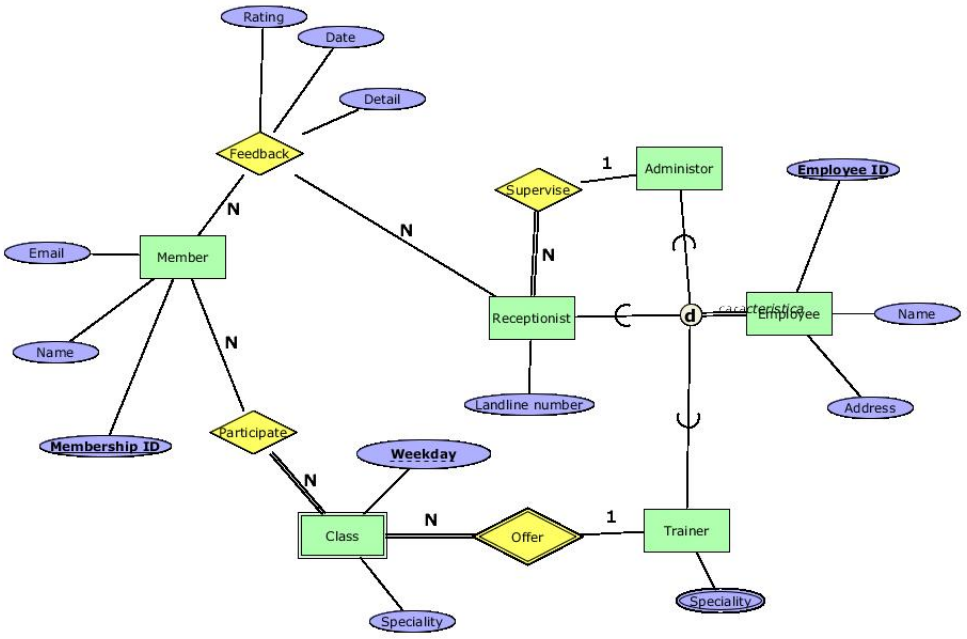
A sports center aims to setup a database to maintain the following information.

The database stores the name, the address and a unique ID of each employee. The employees are classified into three categories: trainers, receptionists and administrators. Each trainer has one or more specialties (e.g., swimming, tennis and squash etc). Each receptionist can be reached through a landline phone number. Each receptionist must be supervised by one administrator and a administrator may supervise multiple receptionists. Each member of this sports center is identified by their membership id and the database also stores their names and emails. Each week a trainer may offer regular training classes that members can participate in. Each training class should focus on one of the specialities of the corresponding trainer and must have a class size no more than 10. Each training class from a trainer is referred to by the day of the week (i.e., Monday, Tuesday, ...) because each trainer offers at most one training class per day. After a member participated in a training class, the member may submit feedback to a receptionist and the feedback contains a rating from 1 (very unsatisfied) to 5 (very satisfied), a date and detailed description.

Your task is to design an Enhanced Entity Relationship (EER) diagram for the above database, which should include entities, relationships, attributes and constraints wherever appropriate (you can make more assumptions if necessary).

You also need to identify the requirements that cannot be captured in an EER-diagram.

Solution:



The requirements that cannot be captured in an EER-diagram:

- A training class should **focus on** one of the specialities of the corresponding trainer and must have a class size no more than 10.
- The feedback contains a rating **from 1 (very unsatisfied) to 5 (very satisfied)**.
- The member may submit feedback to a receptionist **after** participated in a training class.

Common Issues:

- For each weak entity (e.g. for 'class'), the partial key attribute and identifying relationship should be specified.
- For each (normal) entity, a PK should be specified.
- For each binary relationship, cardinality ratios and participation constrains should be specified. Moreover, the cardinality ratio and cardinality limits should not be used together in a single relationship.
- For superclass/subclass, pay attention to the participation and overlap/disjoint constraints.
- Please refer to the feedback file for more specific comments.

Question 2 **3 Marks**

Consider the relation schema R={A, B, C, D, E} and the following set Σ of FDs:

- $A \rightarrow B$
- $AB \rightarrow C$
- $BC \rightarrow AE$
- $DE \rightarrow AB$

2.1 What are the candidate keys of R? Justify your answer (i.e., include the main steps used for finding the candidate keys). (1 Mark)

Solution:

Note that any key must include D as no FD in Σ determines D, so first check D alone:

$$\{D\}^+ = \{D\}$$

Now combinations of 2 attributes with D:

$$\{AD\}^+ = \{ABD\}^+ = \{ABCD\}^+ = \{ABCDE\} \quad ✓$$
$$\{BD\}^+ = \{BD\}$$
$$\{CD\}^+ = \{CD\}$$
$$\{DE\}^+ = \{ABDE\}^+ = \{ABCDE\} \quad ✓$$

The fact that AD and DE are candidate keys means that any remaining candidate keys, if they exist, must be descended from BCD. So check BCD:

$$\{BCD\}^+ = \{ABCDE\} \quad ✓$$

There is no need to check the remaining combinations of 3, 4, or 5 attributes as none of them will be minimal superkeys. Therefore the candidate keys are {AD} , {DE}, and {BCD}.

Common Issues:

- When finding keys, you should justify that a candidate key should be a minimal superkey (not just a superkey), i.e., no redundant attribute can be removed from the candidate keys.

2.2 Find a minimal cover of Σ. Justify your answer (i.e., include the main steps used for finding a minimal cover). (2 Mark)

Solution:

To find the minimal cover we first ensure all FDs only determine one attribute:

$$\Sigma' = \{A \to B, AB \to C, \underline{BC \to A}, \underline{BC \to E}, \underline{DE \to A}, \underline{DE \to B}\}$$

Now we check for redundant attributes on the determining side:

$AB \to C$?	$C \in \{A\}^+$ under Σ? :	$\{A\}^+ = \{AB\}^+ = \{ABC\}^+$	✓	Replace $AB \to C$ with $A \to C$
$BC \to A$?	$A \in \{C\}^+$ under Σ? :	$\{C\}^+ = \{C\}$	✗	B Necessary
$BC \to A$?	$A \in \{B\}^+$ under Σ? :	$\{B\}^+ = \{B\}$	✗	C Necessary
$BC \to E$?	$E \in \{C\}^+$ under Σ? :	$\{C\}^+ = \{C\}$	✗	B Necessary
$BC \to E$?	$E \in \{B\}^+$ under Σ? :	$\{B\}^+ = \{B\}$	✗	C Necessary
$DE \to A$?	$A \in \{E\}^+$ under Σ? :	$\{E\}^+ = \{E\}$	✗	D Necessary
$DE \to A$?	$A \in \{D\}^+$ under Σ? :	$\{D\}^+ = \{D\}$	✗	E Necessary
$DE \to B$?	$B \in \{E\}^+$ under Σ? :	$\{E\}^+ = \{E\}$	✗	D Necessary
$DE \to B$?	$B \in \{D\}^+$ under Σ? :	$\{D\}^+ = \{D\}$	✗	E Necessary

Giving us:

$$\Sigma' = \{A \to B, A \to C, \underline{BC \to A}, \underline{BC \to E}, \underline{DE \to A}, \underline{DE \to B}\}$$

Lastly check if any FDs are redundant:

$A \to B$?	$B \in \{A\}^+$ under $\Sigma - \{A \to B\}$? :	$\{A\}^+ = \{AC\}^+ = \{AC\}$	✗	Keep $A \to B$
$A \to C$?	$C \in \{A\}^+$ under $\Sigma - \{A \to C\}$? :	$\{A\}^+ = \{AB\}^+ = \{AB\}$	✗	Keep $A \to C$
$BC \to A$?	$A \in \{BC\}^+$ under $\Sigma - \{BC \to A\}$? :	$\{BC\}^+ = \{BCE\}^+ = \{BCE\}$	✗	Keep $BC \to A$
$BC \to E$?	$E \in \{BC\}^+$ under $\Sigma - \{BC \to E\}$? :	$\{BC\}^+ = \{ABC\}^+ = \{ABC\}$	✗	Keep $BC \to E$
$DE \to A$?	$A \in \{DE\}^+$ under $\Sigma - \{DE \to A\}$? :	$\{DE\}^+ = \{BDE\}^+ = \{BDE\}$	✗	Keep $DE \to A$
$DE \to B$?	$B \in \{DE\}^+$ under $\Sigma - \{DE \to B\}$? :	$\{DE\}^+ = \{ADE\}^+ = \{ABDE\}$	✓	Remove $DE \to B$

Therefore our minimal cover is (in both minimal Σ_M and canonical Σ_C form):

$$\Sigma_M = \{A \to B, A \to C, BC \to A, BC \to E, DE \to A\}$$
$$\Sigma_C = \{A \to BC, BC \to AE, DE \to A\}$$

Common Issues:

- When applying the minimal cover algorithm, the reasons for removing redundant attributes in LHS or removing redundant FDs should be clarified.

Question 3 **2 Marks**

Consider the relation schema APPOINTMENT={Customer, Branch, Date, Time, Staff, Room} and the following set Σ of FDs:

- Customer, Branch, Date, Time → Staff, Room
- Branch, Date, Time, Room → Customer
- Branch, Date, Room → Staff
- Staff, Date → Branch, Room
- Staff → Branch

Is the above relation schema APPOINTMENT in BCNF? If not, identify a BCNF decomposition for APPOINTMENT. You need to include the main steps used for identifying the BCNF decomposition. Check if this BCNF decomposition is dependency preserving. (2 Mark)

Solution:

We alias the attributes to the first letter of their names for brevity, and order alphabetically.

To check whether Appointment in BCNF, we need check whether all the FDs meet the BCNF requirement (that for all $X \to Y$ in Appointment, X is a super key). We check the closure of the determining side of each FD to see if all the attributes of Appointment are included:

$$\{\text{BCDT}\}^+ = \{\text{BCDSRT}\} \quad ✓$$
$$\{\text{BDTR}\}^+ = \{\text{BCDTR}\}^+ = \{\text{BCDSRT}\} \quad ✓$$
$$\{\text{BDR}\}^+ = \{\text{BDSR}\} \quad ✗$$
$$\{\text{DS}\}^+ = \{\text{BDSR}\} \quad ✗$$
$$\{\text{S}\}^+ = \{\text{BS}\} \quad ✗$$

Hence, Appointment is not in BCNF.

Now we select FDs that violate BCNF requirements and use them to decompose Appointment iteratively. We first select BDR → S and produce the following relations and associated FDs.

$R_A = \{\text{B}, \text{D}, \text{R}, \text{S}\}$ $\quad\quad$ $\Sigma_A = \{\text{BDR} \to \text{S}, \text{DS} \to \text{BR}, \text{S} \to \text{B}\}$

$R_B = \{\text{B}, \text{C}, \text{D}, \text{T}, \text{R}\}$ $\quad\quad$ $\Sigma_B = \{\text{BDTR} \to \text{C}, \text{BCDT} \to \text{R}\}$

Note that BCDT → S is lost.

Note that R_B (with respect to $\Sigma_B = \{\text{BDTR} \to \text{C}, \text{BCDT} \to \text{R}\}$) is now in BCNF because both BDTR and BCDT are superkeys for R_B. However, R_A is not in BCNF because S → B and S is not a superkey in R_A. Thus we select S → B and produce the following decomposition of R_A into R_{A_1} and R_{A_2}.

$R_{A_1} = \{\text{D}, \text{R}, \text{S}\}$ $\quad\quad$ $\Sigma_{A_1} = \{\text{DS} \to \text{R}\}$

$R_{A_2} = \{\text{B}, \text{S}\}$ $\quad\quad$ $\Sigma_{A_2} = \{\text{S} \to \text{B}\}$

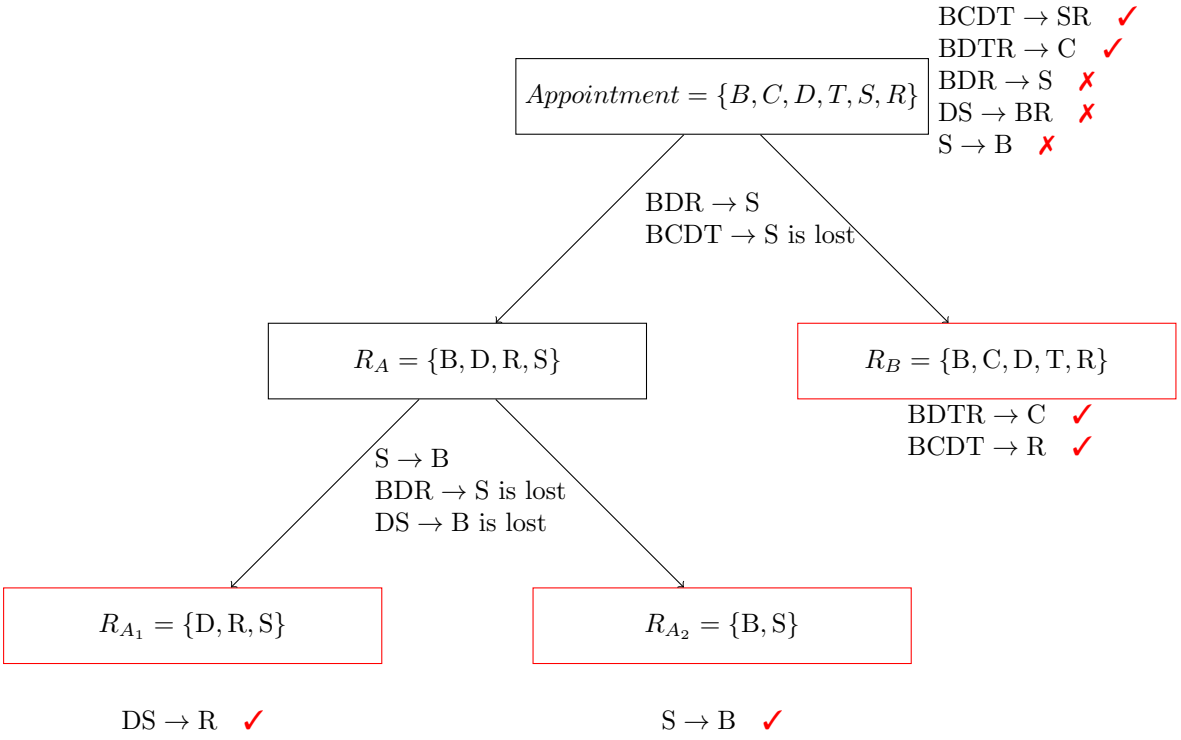
$R_B = \{\text{B}, \text{C}, \text{D}, \text{T}, \text{R}\}$ $\quad\quad$ $\Sigma_B = \{\text{BDTR} \to \text{C}, \text{BCDT} \to \text{R}\}$

Note that BDR → S and DS → B are lost.

Now R_{A_1}, R_{A_2} and R_B are in BCNF and thus $S = \{R_{A_1}, R_{A_2}, R_B\}$ is the BCNF decomposition of Appointment. To verify whether S is dependency-preserving, we need to check whether:

$$\left(\bigcup_{R\in S}\Sigma_R\right) \overset{?}{\to} \{\text{BCDT} \to \text{S}, \text{BDR} \to \text{S}, \text{DS} \to \text{B}\}$$

However, $(\Sigma_{A_1} \cup \Sigma_{A_2} \cup \Sigma_B)$ can not recover the lost FDs. Therefore the FDs cannot be recovered and this decomposition is not dependency-preserving.



NOTE: There may be different ways for BCNF decomposition depending on the order of FDs selected. The above decomposition is just one such possible scenario.

Common Issues:

- When checking the dependency preserving property for BCNF, we need to justify that the lost FDs during decomposition cannot be recovered from surviving ones. (Referring to lecture slides on Page 3, 8-11.)
- In the above example, the decomposition of R_A into R_{A_1} and R_{A_2} is required due to the fact that $\text{S} \to \text{B}$ but S is not a superkey in R_A.

Question 4 **6 Marks**

The following table contains the relational algebra operators covered in our course:

$\sigma_{\varphi}R$	Selection by condition φ
$\pi_{A_1,\ldots,A_n}R$	Projection onto the set of attributes $\{A_1 \ldots, A_n\}$
$\rho_{R'(A_1,\ldots,A_n)}R$	Renaming the relation name to $R^{'}$ and attribute names to $A_1, \ldots, A_n$
$\rho_{R'}R$	Renaming the relation name to $R^{'}$
$\rho_{(A_1,\ldots,A_n)}R$	Renaming the attribute names to $A_1, \ldots, A_n$
$R_1 \cup R_2$	Union of two relations R_1 and R_2
$R_1 \cap R_2$	Intersection of two relations R_1 and R_2
$R_1 - R_2$	Difference of two relations R_1 and R_2
$R_1 \times R_2$	Cartesian product of two relations R_1 and R_2
$R_1 \bowtie_{\varphi} R_2$	Join of two relations R_1 and R_2 with the join condition φ
$R_1 \bowtie R_2$	Natural join of two relations R_1 and R_2
$\varphi_1 \bigwedge \varphi_2$	condition φ_1 AND condition φ_2
$\varphi_1 \bigvee \varphi_2$	condition φ_1 OR condition φ_2

Consider the following relation schemas:

STUDENT={SID, Name, College, Address, Phone} with the primary key {SID},

COURSE={CourseNo, CourseName, Semester} with the primary key {CourseNo, Semester},

LECTURER={LID, Name, College, Email, CourseNo, Semester} with the primary key {LID, CourseNo, Semester} and the foreign key: [CourseNo,Semester]⊆COURSE[CourseNo,Semester],

TUTOR={TID, Email, CourseNo, Semester} with the primary key {TID, CourseNo, Semester} and the foreign keys: [CourseNo,Semester]⊆COURSE[CourseNo,Semester] and [TID]⊆STUDENT[SID],

ENROL={SID, CourseNo, Semester, Unit, Status} with the primary key {SID, CourseNo, Semester} and the foreign keys: [CourseNo,Semester]⊆COURSE[CourseNo,Semester] and [SID]⊆STUDENT[SID].

4.1 Answer the following questions using relational algebra queries. You should only use the relational algebra operators in the above table. You are encouraged to use relational algebra expressions to represent intermediate results if needed. (3 Mark)

[a] List the CourseNo of courses without any tutors in 'S2 2020'? (1 Mark)

Solution:

Select the CourseNo of all the courses in semester 'S2 2020':

$$R_1 = \pi_{CourseNo}(\sigma_{Semester='S22020'}\text{COURSE})$$

Select the CourseNo of all the courses with tutors in semester 'S2 2020':

$$R_2 = \pi_{CourseNo}(\sigma_{Semester='S22020'}(\text{TUTOR})$$

Find the difference: $R_1 - R_2$

[b] List the names of students who have been enrolled in a same course for at least two times (i.e., in different semesters). (1 Mark)

Solution:

Find the students who have been enrolled in a same course for at least two times:

$$R_s = \Big(\rho_{R_1}(\text{ENROL}) \bowtie_{\varphi} \rho_{R_2}(\text{ENROL})\Big)$$

where

$$\varphi = \Big((R_1.CourseNo = R_2.CourseNo) \wedge (R_1.SID = R_2.SID) \wedge (R_1.Semester \neq R_2.Semester)\Big)$$

List the names of students:

$$\pi_{Name}(R_s \bowtie \textsc{Student})$$

[c] List the emails of lecturers who had taught exactly two courses in 'S1 2020'. (1 Mark)

Solution:

$$\textsc{Lecturer'} = \sigma_{Semester='S1\ 2020'}(\textsc{Lecturer})$$

Find the lecturers who had taught at least two courses in 'S1 2020':

$$R_{LEQ2} = \sigma_{\varphi}(\rho_{R_1}(\textsc{Lecturer'}) \times \rho_{R_2}(\textsc{Lecturer'}))$$

where

$$\varphi = \Big((R_1.CourseNo \neq R_2.CourseNo) \wedge (R_1.LID = R_2.LID)\Big)$$

Find the lecturers who had taught at least three courses in 'S1 2020':

$$R_{LEQ3} = \sigma_{\psi}(\rho_{R_1}(\textsc{Lecturer'}) \times \rho_{R_2}(\textsc{Lecturer'}) \times \rho_{R_3}(\textsc{Lecturer'}))$$

where

$$\psi = \Big((R_1.CourseNo \neq R_2.CourseNo) \wedge (R_1.CourseNo \neq R_3.CourseNo) \wedge (R_2.CourseNo \neq R_3.CourseNo)$$

$$\wedge (R_1.LID = R_2.LID) \wedge (R_1.LID = R_3.LID)\Big)$$

List the emails of lecturers using difference:

$$\pi_{R_1.email}(\pi_{R_1.email,R_1.LID}(R_{LEQ2}) - \pi_{R_1.email,R_1.LID}(R_{LEQ3}))$$

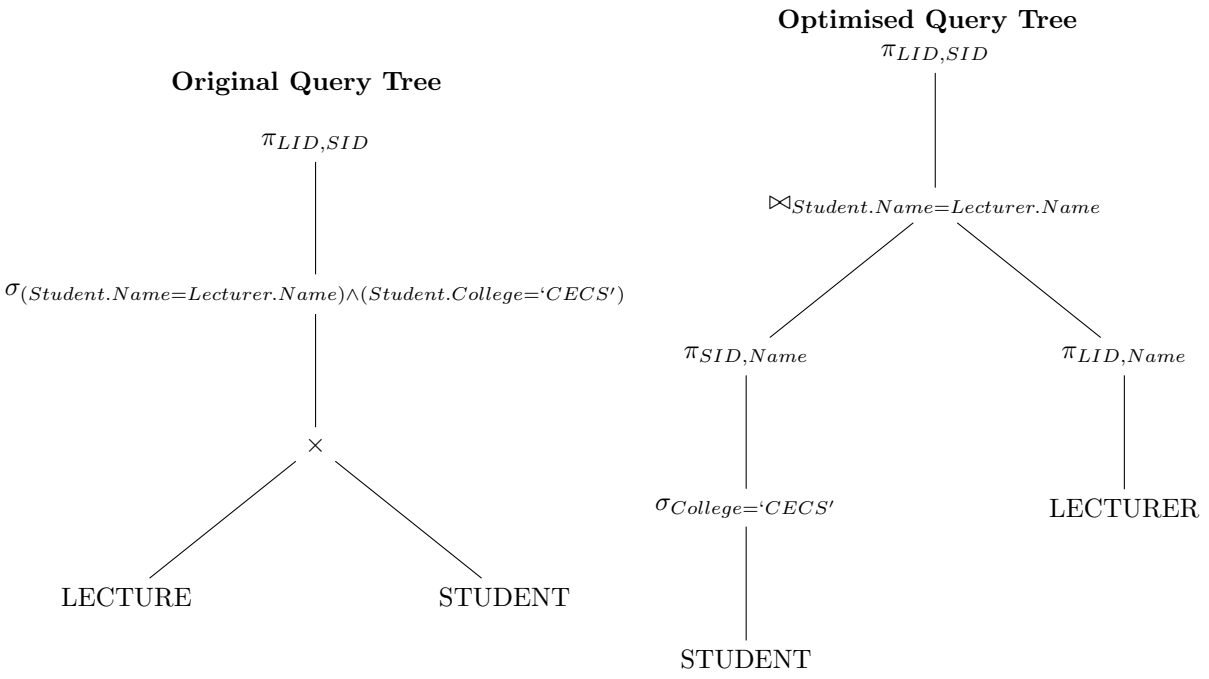
Common Issues:

- As stated in the question specification, you should only use the relational algebra operators in the above table and thus you should not use any aggregation function or count operators.
- When performing self-joining, you need to pay attention to the renaming and joining conditions.

4.2 Optimize the following relational algebra queries (Your marks will depend on how well you present the key ideas of query optimization in your answer). In addition to this, draw the query trees correspond to queries before and after your optimisation. (3 Mark)

[a] $\pi_{LID,SID}(\sigma_{(Student.Name=Lecturer.Name)\wedge(Student.College=`CECS')}(\text{LECTURER} \times \text{STUDENT}))$ (1.5 Mark)

Solution:



The optimised relational algebra is

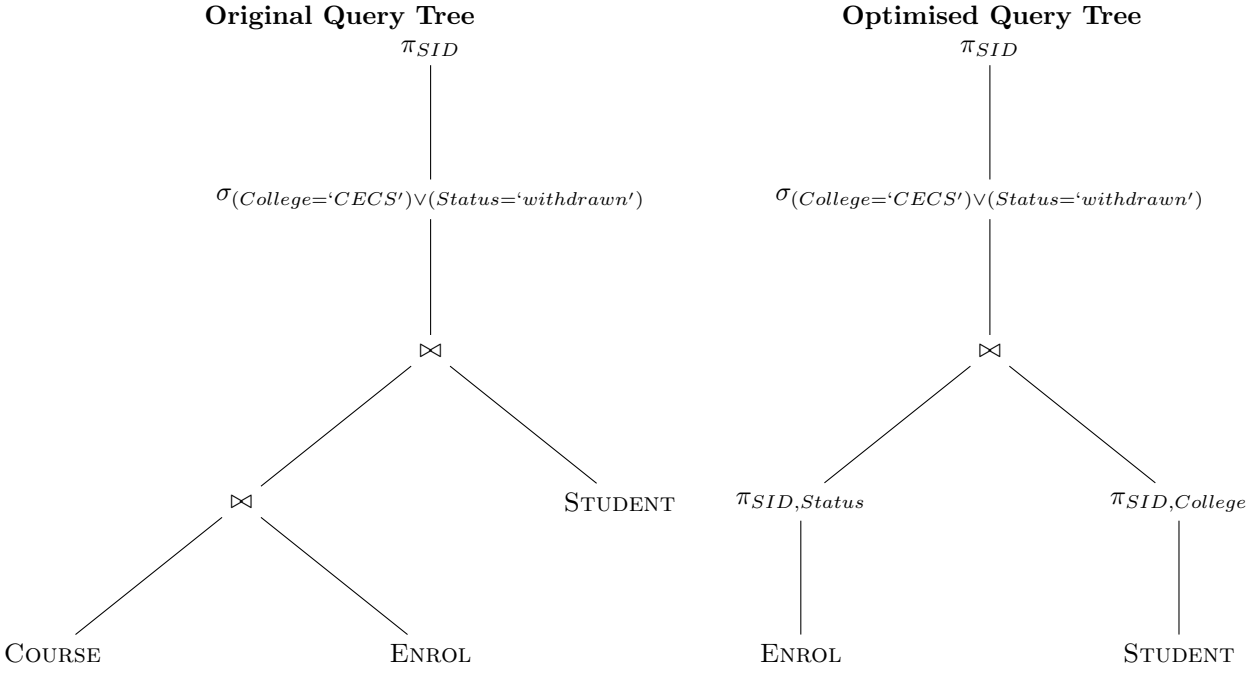
$$\pi_{LID,SID}\left(\left(\pi_{SID,Name}\left(\sigma_{College=`CECS'}(STUDENT)\right)\right) \bowtie_{Student.Name=Lecturer.Name} \left(\pi_{LID,Name}(LECTURER)\right)\right).$$

Common Issues:

- We should push down both projections and selections.

[b] $\pi_{SID}(\sigma_{(College=\text{`}CECS\text{'})\vee(Status=\text{`}withdrawn\text{'})}(\textsc{Course} \bowtie \textsc{Enrol} \bowtie \textsc{Student}))$ (1.5 Mark)

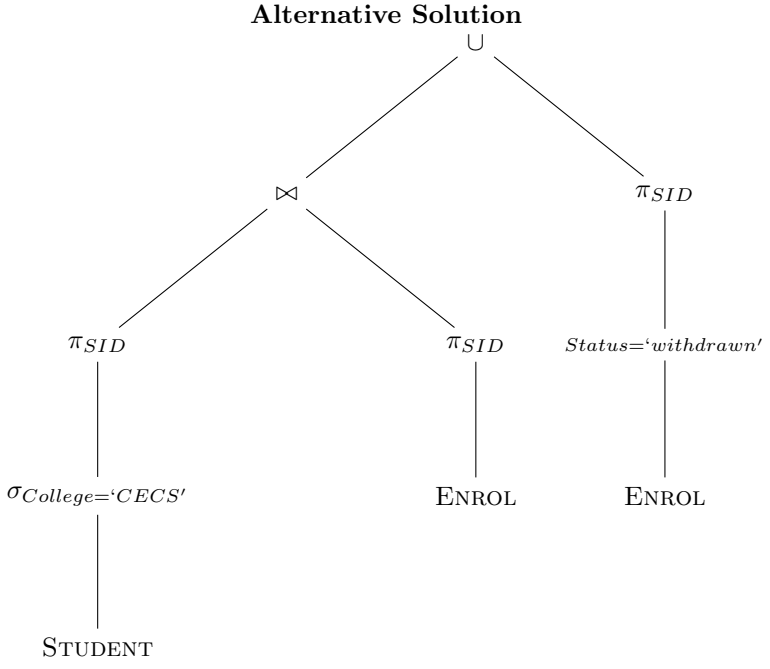
Solution:



The optimised relational algebra is

$$\pi_{SID}\Big(\sigma_{(College=\text{`}CECS\text{'})\vee(Status=\text{`}withdrawn\text{'})}\big((\pi_{SID,Status}(\textsc{Enrol})) \bowtie (\pi_{SID,College}(\textsc{Student}))\big)\Big)$$

Some students proposed the following alternative solution by replacing the 'OR' ($\vee$) by union. Please note that such a solution may not be applicable to more general scenarios.



Common Issues:

- We should use the semantic optimisation to remove the unnecessary join of 'Course'.
- Note two conditions under the selection operator are connected by 'OR' ($\vee$):
  $(College = `CECS') \vee (Status = `withdrawn')$.
  Therefore, the normal push-down selection (with two conditions connected by 'AND') may not work.

+++++