1. **Database design**

For each question in this section, place an **X** beside all answers that apply (there might be more than one answer per question) .

(a) [2 points] Consider the relation R(A,B,C,D,E) with functional dependencies:
$AC \rightarrow B, C \rightarrow A, BD \rightarrow E$
Which of the following functional dependencies hold in this relation?

- ◯ 1. $A \rightarrow C$ ✗
- ☑ 2. $CD \rightarrow E$
- ◯ 3. $C \rightarrow BE$ ✗
- ☑ 4. $EC \rightarrow AB$

$\{A\}^+ = \{A, \times\}$
$\{CD\}^+ = \{CD, AB E\}$
$\{C\}^+ = \{C, AB\}$
$\{EC\}^+ = \{EC, ABE\}$

(b) [2 points] AB, BC, ABC, AC, and C are all the **superkeys** of a relation R(ABC). Which are the **candidate keys** of this relation? Select all that apply.

- ☑ AB
- ◯ ~~BC~~
- ◯ ~~ABC~~ due to C
- ◯ ~~AC~~ due to C
- ☑ C
- ◯ ~~A~~ - Not super key
- ◯ ~~B~~ - Not super key

Candidate need to be Super key and be the min term

(c) [2 points] Given the relation R(A,B,C,D,E), with the set of functional dependencies: $AB \rightarrow D$, $AC \rightarrow E$, $BC \rightarrow D$, $D \rightarrow A$, and $E \rightarrow C$.

Assume we compute the **projection** of this set of functional of dependencies for the attributes **ABC** of R.

Which of the following FDs hold in this projection?

- ◯ $AC \rightarrow B$ ✗ — $\{AC\}^+ = \{AC, E$ 3 - Missing B
- ◯ $A \rightarrow BC$ ✗ — $\{A\}^+ =$ Can't get BC
- ◉ $BC \rightarrow A$ — $\{BC\}^+ = \{BC, DA$ 3 Got A
- ◯ $B \rightarrow AC$ ✗ — Can't get AC

*Need all for Projection to 6 to

(d) [2 points] Assume we have a relation R(A,B,C,D).

Which of the following sets of FDs are **NOT** a minimal basis of this relation?

Select all that apply. Not Canonical

- ✓ ◯ 1. $A \rightarrow BC$, $B \rightarrow C$, $A \rightarrow B$, $AB \rightarrow C$. Extra term
- ✓ ◯ 2. $A \rightarrow B$, $B \rightarrow C$, $AB \rightarrow C$. Not necessary
- ✓ ◯ 3. $A \rightarrow B$, $B \rightarrow C$, $AC \rightarrow D$. Not ? $\{A\}^+ = \{A, BC\}$ so Can Drop AC->D
- ◯ 4. $A \rightarrow B$, $B \rightarrow C$, $DC \rightarrow A$.

2. **Querying a database**

The questions in this section use the Students database (the same we used in class). The schema is the following:

- Relation for students: *S(sid, sname, age, gpa)* with primary key *(sid)*
- Relation for courses: *C(cid, cname, department)* with primary key *(cid)*
- Relation for enrolled: *E(sid, cid, grade)* with primary key *(sid, cid)* with foreign keys:
  - *sid* references S(sid)
  - *cid* references C(cid)

(a) [2 points] Using relational algebra, find the *sid* of students who:

- are taking at least one course from the 'CSC' department (the department of the course is 'CSC')
  or
- their age is bigger than 20

$CSC = \Pi_{C.id}\, \sigma_{dept\ IN\ 'CSC'}\, C$ # Get all CSC courses

$SO = \Pi_{S.id}\, \sigma_{age > 20}\, S$

$CStudent = \Pi_{S.id}\, \sigma_{Cid\ IN\ (CSC)}\, E$

(or) $\Pi_{S.id}\, \sigma_{S.id\ IN\ (CStudent\ OR\ SO)}\, S$

(or) $\Pi_{S.id}\, \sigma_{S.id\ IN\ (CSC)}\, CStudent$

(b) [2 points] Using relational algebra, rewrite this query without using a cross product nor a join (you can use selection, projection and set operations).

$$\Pi_{sid,sname}(S \bowtie E)$$

$\Pi_{S.id, Sname}\left(\sigma_{S.id\ IN\ (\Pi_{S.id}\, E)}\, S\right)$

(c) [2 points] This question uses the schema for the Database students: S for Students, E for Enrolled, and C for courses.

For each pair of Relational Algebra expressions, select it if both queries return the same relation (for any instance of the database). Some of the queries below might be invalid.

○ 1.
- $\Pi_{sid}\sigma_{(cid='csc370' \text{ and } cid='csc330')}E$ — Same Call have 2 diff Vals (empty Rels)
- $(\Pi_{sid}\sigma_{cid='csc370'}E) \cup (\Pi_{sid}\sigma_{cid='csc330'})E$

↑ Diff check ↑

☑ 2.
- $(\Pi_{sid}S) - (\Pi_{sid}E)$ — get all not enrolled
- $\Pi_{sid}\sigma_{(cid \text{ is } null)}(S \overset{\circ}{\bowtie} E)$

↳ not enrolled

☑ 3.
- $A = \Pi_{sid}E$ — get s:d of all enrolled
  $\Pi_{sid,sname}\sigma_{(sid \text{ in } A)}S$ — get name/id of all enrolled
- $\Pi_{sid,sname}(S \bowtie^{R} E)$ — get all enrolled

○ 4.
- $E \overset{\circ}{\bowtie} E$? They should Be ?
- E

(d) [2 points] Write a relational algebra expression to find the *sid* of the student (or students, there might be more than one) with the highest GPA. (Use only the relational algebra operators we have covered in class).

Hint: One method involves finding the sid of students who do not have the highest GPA.

* Need to loop over for check

$S = S1 = S2$

$\Pi_{S.sid} S - \Pi_{S.sid} \sigma_{S1.gpa < S2.gpa \text{ and } S1.sid = S2.sid} (S1 \times S2)$

(all students) — Remove — (all lowest)

OR

$\Pi_{sid} \sigma_{gpa = All} (\Pi_{gpa} S)^{S}$

↑ get max ↑ To remove