CITY UNIVERSITY OF HONG KONG

Course code & title : CS3402 Database Systems

Session : Semester B 2019/20

Time allowed : 1.5 hours

---

1. This paper consists of SIX questions (including this cover page).

2. Write down your answer by hand on paper (A4 size, white background, no grid lines), and write with black pen. No pencil accepted.

---

*This is an **open-book** examination.*

Candidates are allowed to use the following materials/aids:

Lecture/tutorial notes, personal notes, and textbook.

Candidates will be subject to disciplinary action if any unauthorized materials or aids are found on them.

No network searching, and no discussion during the exam.

If suspiciously similar answers are found, you may be subject to disciplinary action. We strongly recommend that you videotape yourself while taking the exam. You can keep the video as evidence of academic honesty and as evidence of an earnest attempt to complete the assessment, should technical issues prevent you from submitting your answers.

Q1 (20%)	Q2 (30%)	Q3 (15%)	Q4 (18%)	Q5 (17%)	Total (100%)

Problem One: Basic Concepts (20 points)
1) Judge whether each of the following statement is TRUE or FALSE and give your justification for your answer. [15 points]

//Marking tips: Correct answer is worth 2 points; correct explanation 1 point!
(a) Database System is a set of software programs for creating, storing, updating and accessing the data of a database. Oracle, MySQL, Microsoft SQL Server, DB2 and Microsoft Access are all Database Systems.
Answer: False. Database Manage System (DBMS) is a set of software programs for creating, storing, updating and accessing the data of a DB; while Database System is an integrated system of hardware, software, people, procedures, and data.

(b) If a 3NF table has two candidate keys and two candidates have no overlapping attribute, then this table is also in BCNF.

Answer: True. BCNF is different from 3NF only when the table has two or more candidate keys which are overlapping.

(c) The SQL statement SELECT * FROM COURSE_GRADE WHERE GPA < 3.0 OR GPA >= 3.0 is equivalent to SELECT * FROM COURSE_GRADE WHERE GPA != NULL.

Answer: False. The second one is wrong, must use IS NULL.

(d) If $U \div S = T$, U must be equal to $S \times T$

Answer: False. $S \times T$ is a subset of U.

(e) Since View is the virtual table, if we change the value of tuples in view, the database state is not changed.

Answer. False. Changes in view will be reflected in the base table, thus the database state is updated.

2 List five major advantages of DBMS compared to traditional file systems. [5 Points]

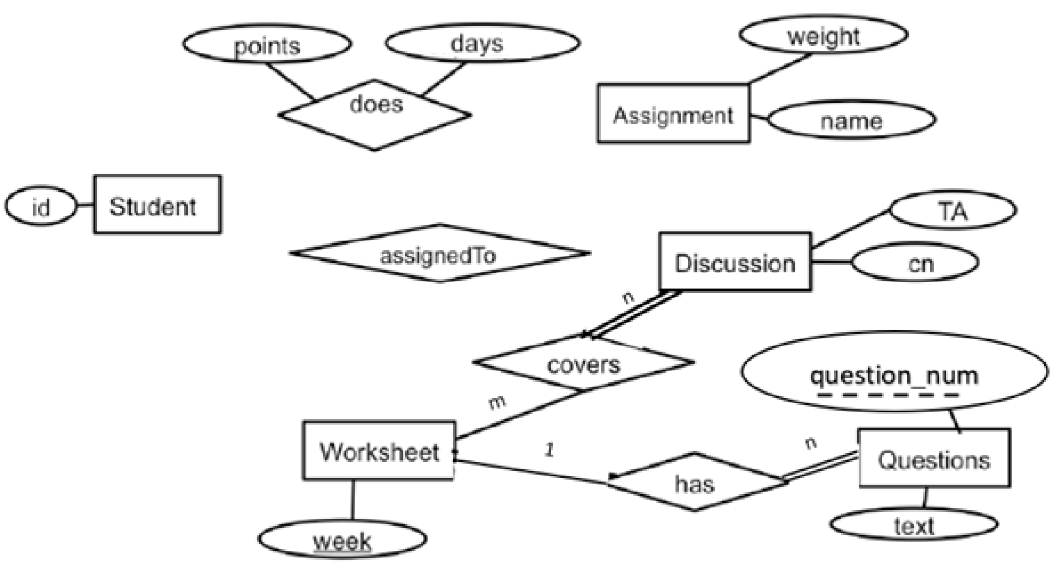
Answer:
Controlling Redundancy, Enforcing Integrity Constraints, Restricting Unauthorized Access, Representing Complex Relationships among Data, Providing Storage Structures and Search Techniques for Efficient Query Processing, Providing Backup and Recovery, Providing

Persistent Storage for Program Objects, Providing Multiple User Interfaces and concurrent control, etc

Problem Two: ER Model (30 points)
We are going to design a database for the CS3402 course to keep track of the information about students who are taking this course.

- In this course, each student can be assigned to one optional group discussion section with more than one participant. Each group discussion section has a unique course number (cn) and is hosted by a TA.
- Students will also do some assignments with different weights. For each assignment, we want to store the number of days the student has spent and the points the student earned on the assignment. If a student does not turn in an assignment, he/she will be recorded in the database with zero points. Assume there are many assignments, and all assignments are compulsory.

1. To implement the database, we first need an ER diagram. Part of the ER diagram is shown below. Please complete the ER diagram by following the steps below: [12 points]

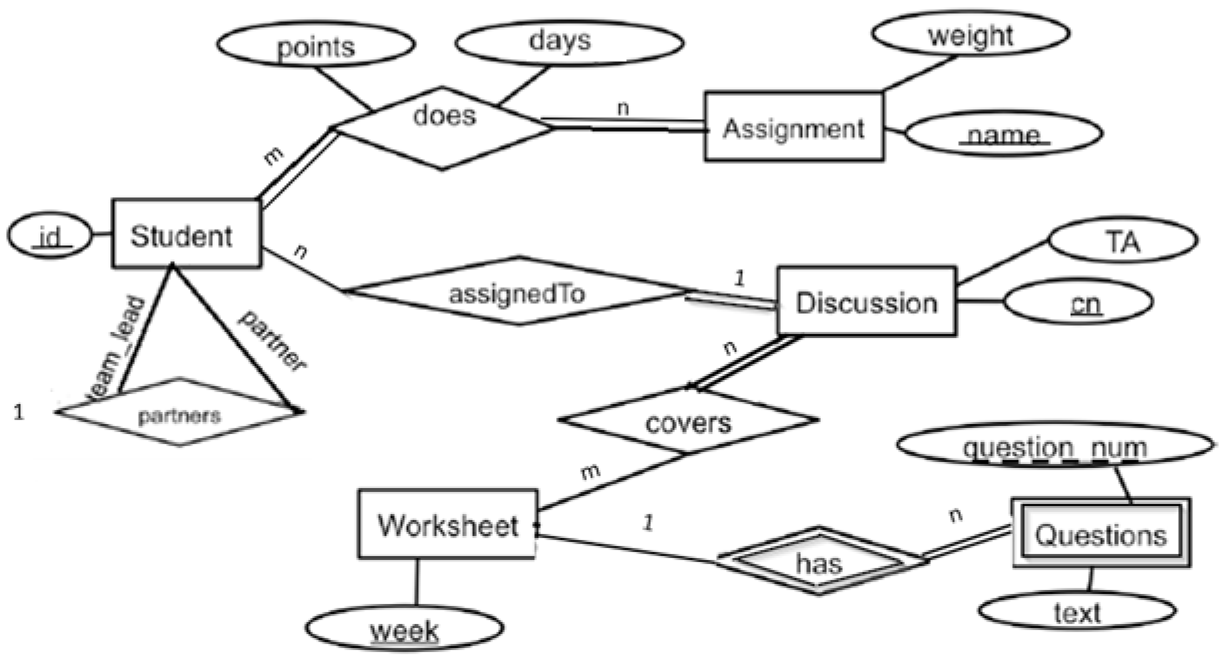


(a) Complete the ER Diagram for the Student, Assignment, and Discussion entities by underlining the primary keys, connecting the given entity and relationship sets, and indicating the cardinality and participation constraints.

(b) Proper notation for the weak entity is missing in the given ER diagram. Please complete the notation for the weak entity.

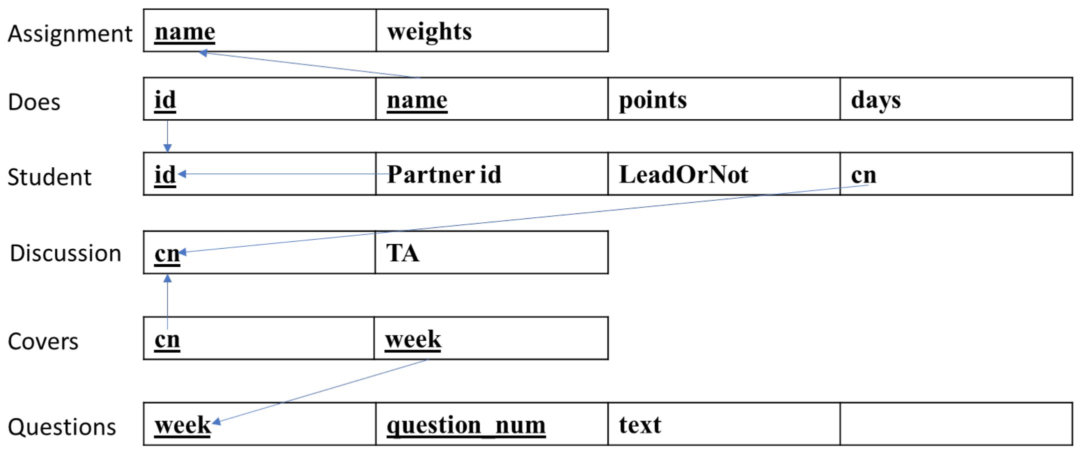
(c) Consider the following restriction: all students in this course must work in teams of two. Each team should have one student as the lead. Assume that the number of students is even. Please make the additions to the ER diagram to do so. You should not add more than one entity and/or relationship.

Answer:

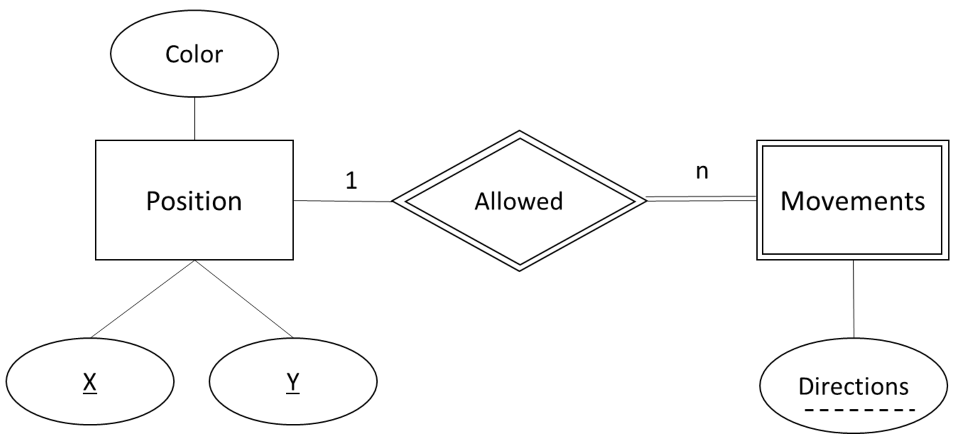


2. Convert the completed ER diagram to the Relational Model. [12 points]

Answer:



3. Consider the following ER diagram, which captures the allowed movements of a game character at each position of a map:



Please implement this database by creating tables with SQL statements while maintaining the constraints defined in the ER diagram. Assume that X and Y are integers; Color and Direction are strings of at most 32 characters. [6 points]

Answer:

```
CREATE TABLE POSITION
    (Color VARCHAR(32) Not NULL,
     X INTEGER,
     Y INTEGER,
     Primary Key (X, Y)
   );

CREATE TABLE MOVEMENTS
    (Direction VARCHAR(32),
```

```
  X INTEGER,
  Y INTEGER,
  Primary Key (Direction, X, Y),
  Foreign Key (X) References Position(X),
  Foreign Key (Y) References Position(Y)
);
```

Problem Three: Integrity Constraints (15 points)
Suppose we have a relational database containing two tables DEPT(DEPTNO,DNAME,LOC) and EMP(EMPNO,ENAME,JOB,MGR,HIREDATE,SAL,COMM,DEPTNO). Two foreign keys are defined in EMP: EMP.DEPTNO referencing DEPT.DEPTNO and EMP.MGR referencing EMP.EMPNO. The current state of the database is shown as the figure below.

DEPT

DEPTNO	DNAME	LOC
10	ACCOUNTING	NEW YORK
20	RESEARCH	DALLAS
30	SALES	CHICAGO
40	OPERATIONS	BOSTON

EMP

EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO
7369	SMITH	CLERK	7902	17-DEC-80	800		20
7499	ALLEN	SALESMAN	7698	20-FEB-81	1600	300	30
7521	WARD	SALESMAN	7698	22-FEB-81	1250	500	30
7655	JONES	MANAGER	7839	2-APR-81	2975		20
7654	MARTIN	SALESMAN	7698	28-SEP-81	1250	1400	30
7698	BLAKE	MANAGER	7839	1-MAY-91	2850		30
7782	CLARK	MANAGER	7839	9-JUN-81	2450		10
7788	SCOTT	ANALYST	7655	21-MAR-87	3000		20
7839	KING	PRESIDENT		12-NOV-81	5000	0	10
7844	TURNER	SALESMAN	7698	18-SEP-81	1500		30
7876	ADAMS	CLERK	7788	24-APR-87	1100		20
7900	JAMES	CLERK	7698	3-DEC-81	950		30
7902	FORD	ANALYST	7655	3-DEC-81	3000		20
7934	MILLER	CLERK	7782	3-JAN-81	1300		10

Suppose each of the following Update operations is applied directly to the database. Discuss all integrity constraints violated by each operation, if any, and the different ways of enforcing these constraints.

1. INSERT INTO EMP VALUES (7698, 'BLACK', 'CLERK', 7901, '16-DEC-85', 1800, NULL, 20); [5 points]

Answer: Violates both the key constraint and referential integrity constraint.

Violates the key constraint because there already exists an EMP tuple with EMPNO=4. We may enforce this constraint by: (i) rejecting the insertion, or (ii) changing the value of EMPNO in the new EMP tuple to a value that does not already exist.

Violates referential integrity because MGR='7901' and there is no tuple in the EMP relation with EMPNO='7901'. We may enforce the constraint by: (i) rejecting the insertion, (ii) changing the value of MGR to an existing EMPNO value in EMP, or (iii) inserting a new EMP tuple with EMPNO ='7901'.

2. Delete DEPT where DANME= 'SALES'; [5 points]

Answer: Violates referential integrity constraint because several tuples exist in the EMP relation that reference the tuple being deleted from DEPT. We may enforce the constraint by: (i) rejecting the deletion, or (ii) deleting all tuples in the EMP relations whose values for DEPTNO is equal to 30.

3. Update EMPNO=NULL, MGR=NULL where EANME= 'JONES' [5 points]

Answer: Violates both entity integrity and referential integrity constraints.
Violates the entity integrity constraint because EMPNO is the key attribute which does not allow to be NULL.

Violates referential integrity because several tuples exist in the EMP relation that their MGR referencing to the tuple with EMPNO=7655.

We may enforce the constraint by: (i) rejecting the update, or (ii) changing the value of EMPNO to a value that is not null and does not already exist in the table. At the same time, we need to update the MGR values to the new value for all EMP tuples where MGR=7655.

Problem Four: SQL and Relational Algebra (18 points)
Given the following relations containing students' grade records:

Student (**<u>StudentID</u>**: *integer*, **Name**: *string*, **Year**: *date*)
Course (**<u>CourseID</u>**: *integer*, **Title**: *string*, **InstructorName**: *string*, **Office**: *string*)
GradeRecord (**<u>StudentID</u>**: *integer*, **<u>CourseID</u>**: *integer*, **Grade**: *string*, **Score**: *float*)

Note that there are no duplicate records in the three relations.
Write the following queries in **BOTH** relational algebra and SQL.

1. Find the **Name**s of students who got grade "A" with a score of more than 85 in at least one course. [6 points]

Answer:
$stuID \leftarrow \pi_{StudentID} (\sigma_{Grade='A' \text{ and } Score>85} \text{ (Record) })$
π_{Name} (stuID * Student)

Select Name from Student

where StudentID in (SELECT distinct StudentID FROM Record WHERE Grade='A' AND Score > 85);

2. Find the **Names** of students who have taken both courses "dbs"(CourseID:3402) and course "c++"(CourseID: 1002). [6 points]

Answer:

$DBS \leftarrow \sigma_{CourseID=3402}(Record)$

$CPP \leftarrow \sigma_{CourseID=1002}(Record)$

$stuID \leftarrow \pi_{StudentID}(DBS \bowtie_{DBS.StudentID=CPP.StudentID} CPP)$

$\pi_{Name}(stuID * Student)$

Select Name from Student where StudentID IN
(SELECT Distinct DBS.StudentID FROM GradeRecord AS DBS where WHERE CourseID=3402)
INTERSECT
(SELECT Distinct CPP.StudentID FROM GradeRecord AS CPP where WHERE CourseID=1002) ;

3. Find the **StudentIDs** of students who took all courses taught by instructor "James". [6 points]

Answer:

$JamesCourse \leftarrow \pi_{CourseID}(\sigma_{InstructorName = "James"}(Course))$

$\pi_{StudentID, CourseID} GradeRecord \div JamesCourse$

SELECT StudentID from Stundent AS X
where NOT EXISTS
 (SELECT * FROM Course AS Y WHERE Y.InstructorName = "James" and NOT EXISTS
 (SELECT * FROM GradeRecord AS Z WHERE Z.StudentID=X.StudentID and Z.Coursed = Y.CourseID)
);

Problem Five: Normalization [17 points]

1. Consider a relation R(A,B,C) and suppose R contains the following four tuples:

A (Position)	B (Name)	C (Nice)
TA	Xiangyi MENG	yes
TA	Xinyu WANG	yes
TA	Zhengxiang ZHOU	yes
Teacher	Jialu HUANG	yes

For each of the following functional dependencies, state whether each functional dependency is satisfied [satisfied/not satisfitd] by this relation instance. [4 points]

(a) A → B
(b) C → A
(c) AB → C
(d) AC → B

Answer:

(a) not satisfied
(c) not satisfied
(c) satisfied
(d) not satisfied

2 Suppose we have a relational database table R for schedules of courses and sections at a university, consisting of the following attributes:
CN(CourseNo), SN(SectionNo), OD(OfferingDept), CR(Credit), CL(CourseLevel), IN(InstructorName), SM(Semster), Y(Year), DH(DayHour), RN(RoomNo), NS(NumbersOfStudents) with the following functional dependencies:
{CN → {OD, CR, CL}, {CN,SN,SM,Y} →{DH,RN,NS,IN},{RN,DH,SM,Y}→{IN,CN,SN}}.

(a) Prove that {RN,DH,SM,Y }→CL holds by using the inference rules. [3 points]

Answer: {RN,DH,SM,Y }→{IN,CN,SN} (Given) (1)

From (1), we have {RN,DH,SM,Y }→CN (decomposition rule) (2)

CN → {OD, CR, CL} (Given) (3)

From (2)(3), we have {RN,DH,SM,Y }→{OD, CR, CL} (transitive rule) (4)

Form (4), we have {RN,DH,SM,Y }→ CL (decomposition rule)

(b) Find all candidate keys of this database by computing the closures. [5 points]
Answer: Both K1 = { CN,SN,SM,Y } and K2 = { RN,DH,SM,Y } are (candidate) keys of R, because $\{ CN,SN,SM,Y \}^{+}$={ CN,SN,SM,Y, RN,DH,SM,Y, OD, CR, CL } and $\{ RN,DH,SM,Y \}^{+}$={ CN,SN,SM,Y, RN,DH,SM,Y, OD, CR, CL }, and removing any attribute from K1 or K2, its closure cannot cover all attributes.

(c) Normalize this database into BCNF. [5 points]
Answer: Applying the general definition of 2NF, we find that the functional dependency {CN} →{OD, CR, CL} is a partial dependency for K1 (since C is included in K1). Hence, R is normalized into R1 and R2 as follows:
R1 = {<u>CN</u>, OD, CR, CL} (2 points)
R2 = {<u>RN, DH, SM, Y</u>, IN, CN, SN, NS} (or R2 = {RN, DH, SM, <u>Y</u>, IN, <u>CN, SN, NS</u>}) with candidate keys K1 and K2 (2 points)
Since neither R1 nor R2 have transitive dependencies on either of the candidate keys, R1 and R2 are in 3NF also. They also both satisfy the definition of BCNF. (2 points)