



CZ2007 Introduction to Databases

Tutorial 7 SQL



Question 1(a)

- (1) Find the name(s) of student(s) with the lowest GPA.

STUDENT(Sid, Sname, Sex, Age, Year, GPA)
DEPT(Dname, Numphds)
PROF(Pname, Dname)
COURSE(Cno, Cname, Dname)
MAJOR(Dname, Sid)
SECTION(Dname, Cno, Sectno, Pname)
ENROLL(Sid, Grade, Dname, Cno, Sectno)

```
SELECT      Sname
FROM        STUDENT
WHERE       GPA = (SELECT MIN(GPA) FROM STUDENT);
```

“=” in the **WHERE** condition here could be “IN”



Question 1 (b)

- (2) Consider the relation R (A,B,C,D) with candidate keys AC and D.
- What will be the output of the following query? Justify your answer.

```
SELECT  A, B
FROM    R
WHERE   C > (SELECT D FROM R WHERE C = 3);
```

This query will report error.

- The subquery does not return a scalar, and therefore cannot be compared against a single data value. You would need to add “ANY” or “ALL” before the subquery for the query to run.
- Columns C and D may not be of the same data type. Hence it is not clear whether $C > \text{Any/All} (D)$ would be meaningful, even if “ANY” or “ALL” are added.



Question 2

➤ Consider the following relational schema

Reader (RDNR, Surname, Firstname, City, Birthdate)
Book (ISBN, Title, Author, NoPages, PubYear, PublisherName)
Publisher (PublisherName, PublisherCity)
Category (CategoryName, Belongsto)
Copy (ISBN, CopyNumber, Shelf, Position)
Loan (ReaderNr, ISBN, Copy, ReturnDate)
BookCategory (ISBN, CategoryName)

- BelongsTo refers to which parent categories the current category belongs to.
- Each book has a specific ISBN, and many copies of a book might be available under the same ISBN.
- A reader may borrow the same copy for multiple times, and each instance is recorded by its ReturnDate.
- All the categories that a book belongs to are stored in the table BookCategory.

Question 2 (a)

➤ Consider the following relational schema

Reader (RDNR, Surname, Firstname, City, Birthdate)
Book (ISBN, Title, Author, NoPages, PubYear, PublisherName)
Publisher (PublisherName, PublisherCity)
Category (CategoryName, BelongsTo)
Copy (ISBN, CopyNumber, Shelf, Position)
Loan (ReaderNr, ISBN, Copy, ReturnDate)
BookCategory (ISBN, CategoryName)

➤ (a) Which categories do not have any subcategories.

```
SELECT      C1.CategoryName      FROM Category C1
WHERE NOT EXISTS
(SELECT      C2.CategoryName      FROM Category C2
WHERE      C2.BelongsTo=C1.CategoryName);
```

Question 2 (b)

Reader (RDNR, Surname, Firstname, City, Birthdate)
Book (ISBN, Title, Author, NoPages, PubYear, PublisherName)
Publisher (PublisherName, PublisherCity)
Category (CategoryName, BelongsTo)
Copy (ISBN, CopyNumber, Shelf, Position)
Loan (ReaderNr, ISBN, Copy, ReturnDate)
BookCategory (ISBN, CategoryName)

- (b) For which of the books there is at least one copy available

```
SELECT Title FROM Book WHERE ISBN IN (  
    SELECT ISBN FROM (  
        (SELECT CopyNumber, ISBN FROM Copy)  
        EXCEPT  
        (SELECT Copy, ISBN FROM Loan))));
```

```
SELECT ISBN, Title FROM Book WHERE ISBN IN (  
    SELECT ISBN FROM (  
        (SELECT CopyNumber, ISBN FROM Copy)  
        EXCEPT  
        (SELECT Copy, ISBN FROM Loan WHERE ReturnDate > CURDATE())));
```



Question 2 (c)

- Which books have more pages than twice the average of the number of pages of all books

Reader (RDNR, Surname, Firstname, City, Birthdate)
Book (ISBN, Title, Author, NoPages, PubYear, PublisherName)
Publisher (PublisherName, PublisherCity)
Category (CategoryName, BelongsTo)
Copy (ISBN, CopyNumber, Shelf, Position)
Loan (ReaderNr, ISBN, Copy, ReturnDate)
BookCategory (ISBN, CategoryName)

```
SELECT ISBN, Title FROM Book
WHERE NumberOfPages >= 2* (
    SELECT AVG(NumberOfPages) FROM Book);
```

Question 2 (d)

➤ What are the surnames of the readers from the city “New York”

Reader (RDNR, Surname, Firstname, City, Birthdate)
Book (ISBN, Title, Author, NoPages, PubYear, PublisherName)
Publisher (PublisherName, PublisherCity)
Category (CategoryName, BelongsTo)
Copy (ISBN, CopyNumber, Shelf, Position)
Loan (ReaderNr, ISBN, Copy, ReturnDate)
BookCategory (ISBN, CategoryName)

```
SELECT DISTINCT Surname FROM Reader  
WHERE City = 'New York'
```


Question 3

- For the following relational schema
 - Employee (employee-name, street, city)
 - Works (employee-name, company-name, salary)
 - Company (company-name, city)
 - Manages (employee-name, manager-name)
- Find the names of all employees who earn more than the average salary of all employees of their company.
- Assume that all people work for at most one company.

```
SELECT employee-name FROM Works t
WHERE salary > (
    SELECT AVG (salary) from Works s
    WHERE t.company-name = s.company-name)
```

Question 4

- Suppose we are maintaining a database of articles published in our newspaper, the Straits Times. We have the following schema (where keys are underlined):
- Article (issueID, articleID, author, title)
 - Citation (articleID, issueID, citedArticleID, citedIssueID)
 - WordAppears (wordID, issueID, articleID, position)
 - Words (wordID, wordText)
 - Issue (issueID, date, howManyDistributed)
- Assume that dates can be compared using comparison operators ($<$, $>$, $=$). Assume that position is an index specifying where the word appears (1 = first word, 2 = second, etc.).

Question 4 (ii)

- Find the most cited articles in the newspaper history

Article (issueID, articleID, author, title)
Citation (articleID, issueID, citedArticleID, citedIssueID)
WordAppears (wordID, issueID, articleID, position)
Words (wordID, wordText)
Issue (issueID, date, howManyDistributed)

```
SELECT      citedIssueID, citedArticleID
FROM        Citation
GROUP BY    citedIssueID, citedArticleID
HAVING COUNT(*) >= ALL
              (SELECT COUNT(*)
               FROM      Citation C1
               GROUP BY   C1.citedArticleID, C1.citedIssueID);
```

Question 4 (ii)

- Find the number of citations/**references** per author for “senior” authors (i.e., an author who has at least one article that was published 10 or more years ago)

Article (issueID, articleID, author, title)

Citation (articleID, issueID, citedArticleID, citedIssueID)

WordAppears (wordID, issueID, articleID, position)

Words (wordID, wordText)

Issue (issueID, date, howManyDistributed)

```
SELECT      a.author, COUNT(*)
FROM        Article a, Citation c
WHERE       a.issueID = c.citedIssueID AND a.articleID = c.citedArticleID
AND EXISTS
    (SELECT *
     FROM    Article a2, Issue i
     WHERE   a2.issueID = i.issueID AND a2.author = a.author AND
            Year(getDate()) - Year(i.date) >= 10)
GROUP BY   a.author;
```