# Sample DB Examination Solutions (Spring, 2007)

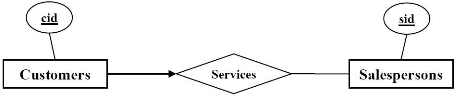
May 29, 2007

# Question 1:

Draw a simple ER diagram that results in a primary key / foreign key constraint to be created between the tables:

```
CREATE TABLE Salespersons(
    sid CHAR(10), primary key (sid))

CREATE TABLE Customers(
    cid CHAR(10), sid CHAR(10),
    primary key (cid),
    foreign key (sid) reference Salespersons)
```



# Question: 2

Express query "*Find the* **passenger_name** *of all passengers who have a seat booked on at least one plane of* **every** *type*" in relational algebra

FLIGHTS(*flight_num*, source_city, destination_city)
DEPARTURES(*flight_num*, *data*, plane_type)
PASSENGERS(*passenger_id*, passenger_name, passenger_address)
BOOKINGS(*passenger_id*, *flight_num*, *date*, seat_number)

$$\pi_{passenger\_name, plane\_type}(PASSENGERS \bowtie BOOKINGS \bowtie DEPARTURES) / \pi_{plane\_type}(DEPARTURES)$$

# Question 3:

Express query "*Find the* **passenger_id** *of all passengers who have a seat booked on a plane of type "747" from San Francisco to Washington.*"

Do not return any duplicate values

```
SELECT DISTINCT B.passenger_id from FLIGHTS F, DEPARTURES D, BOOKINGS B
    WHERE B.flight_num = D.flight_num
    AND B.date = D.date
    AND F.flight_num = D.flight_num
    AND F.source_city = 'San Francisco'
    AND F.destination_city = 'Washington'
    AND D.plane_type = '747'
```

# Question 4:

Based on the following given SQL queries, which answer is correct:

$Q1$ :

SELECT a
FROM R
WHERE b >= **ANY**(SELECT d FROM S WHERE c > 10);

$Q2$ :

SELECT a
FROM R
WHERE b >= **ALL**(SELECT d FROM S WHERE c > 10);

(d) Q1 and Q2 produce different answers

# Question 5:

Which answer is correct for the schema of relation R

*Q*1 :

```
SELECT x
FROM R rr
WHERE NOT EXISTS(
    SELECT * FROM R WHERE x > rr.x);
```

*Q*2 :

```
SELECT MAX(x) FROM R;
```

(c) The answer to Q2 is always contained in the answer to Q1

# Question 6:

Which statement about the relation R(a,b) and the following SQL query is true:

```
SELECT a, MAX(b), MIN(b)
FROM R
GROUP BY a;
```

relational algebra

$\rho(R_1(a \rightarrow c, b \rightarrow d), R)$

$\rho(S1, \pi_{a,b}(\sigma_{b \geq d}(R \times R1)))$

$\rho(S2(b \rightarrow c), \pi_{a,b}(\sigma_{b \leq d}(R \times R1)))$

$\rho(S3, R - S1)$

$\rho(S4, R - S2)$

$Answer = S3 \bowtie S4$

(d) This query can be written in relational algebra

# Schedule-1:

| | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| T1 | R(A) | R(C) | | | W(C) | | | | | | commit |
| T2 | | | R(C) | | | | W(C) | R(B) | W(B) | | commit |
| T3 | | | | R(C) | | R(A) | | | | W(A) | commit |

- Not serializable
- Non conflict-serializable

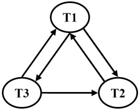



Figure: Precedence Graph

# Schedule-2:

| | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| T1 | | R(A) | | R(C) | W(A) | commit | | | | | |
| T2 | | | R(C) | | | | R(B) | W(B) | commit | | |
| T3 | R(B) | | | | | | | | | W(B) | commit |

- Not serializable
- Non conflict-serializable

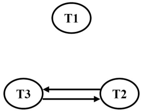



Figure: Precedence Graph

# Schedule-3:

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| T1 | R(C) | W(A) | | | W(A) | commit | |
| T2 | | | W(A) | R(B) | | W(B) | commit |

- Serializable as $T_1 \rightarrow T_2$ (Blind Write)
- Non conflict-serializable

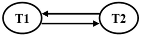



Figure: Precedence Graph

# Strict 2PL on Schedule 1:

- Deadlock happen

**Table: Strict 2PL on Schedule 1**

| | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| $T_1$ | S(A) | R(A) | S(C) | R(C) | | | | | Wait $T_2$, $T_3$ | | | | |
| $T_2$ | | | | | S(C) | R(C) | | | | | | Wait $T_1$, $T_3$ | |
| $T_3$ | | | | | | | S(C) | R(C) | | S(A) | R(A) | | Wait $T_1$ |

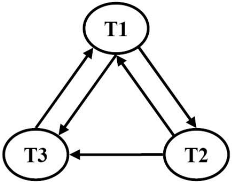



**Figure: Wait-for Graph on Schedule 1**

# Strict 2PL on Schedule 2:

- Deadlock happen

**Table: Strict 2PL on Schedule 2**

| | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| $T_1$ | | | S(A) | R(A) | | | S(C) | R(C) | X(A) | W(A) | CO | | | | |
| $T_2$ | | | | | S(C) | R(C) | | | | | | S(B) | R(B) | Wait $T_3$ | |
| $T_3$ | S(B) | R(B) | | | | | | | | | | | | | Wait $T_2$ |

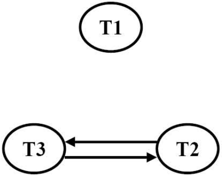



**Figure: Wait-for Graph on Schedule 2**

# Strict 2PL on Schedule 3:

- No Deadlock

**Table: Strict 2PL on Schedule 3**

| | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| $T_1$ | S(C) | R(C) | X(A) | W(A) | | W(A) | CO | | | | | | | |
| $T_2$ | | | | | Wait $T_1$ | | | X(A) | W(A) | S(B) | R(B) | X(B) | W(B) | CO |

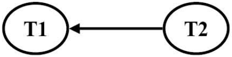



**Figure: Wait-for Graph on Schedule 3**

# Question 1:

Consider a relational schema $\underline{A}BCDEFGHIJ$, which contains the following FDs:

$$AB \rightarrow C, D \rightarrow E, AE \rightarrow G, GD \rightarrow H, IF \rightarrow J$$

Task 1: $(a) ABD \rightarrow GH, (b) ABD \rightarrow HJ, (c) ABC \rightarrow G, (d) GD \rightarrow HE$

(a) $AD \rightarrow AE(Augementation : D \rightarrow E); AD \rightarrow G(Transitivity : AD \rightarrow AE, AE \rightarrow G);$
$AED \rightarrow GD(Augementation : AE \rightarrow G); AED \rightarrow H(Transitivity : AED \rightarrow GD, GD \rightarrow H);$
$AD \rightarrow H(AED \rightarrow H, D \rightarrow E); AD \rightarrow GH(Union : AD \rightarrow G, AD \rightarrow H);$
$ADB \rightarrow GHB(Augementation : AD \rightarrow GH); ABD \rightarrow GH(Decomposition : ABD \rightarrow GHB)$

(d) $GD \rightarrow D(Reflixivity : D \subseteq GD); GD \rightarrow E(Transitivity : GD \rightarrow D, D \rightarrow E)$
$GD \rightarrow HE(Union : GD \rightarrow H, GD \rightarrow E)$

Task 4: Any relation that satisfies $X \rightarrow Y$ and $X \rightarrow Z$ must also satisfy $X \rightarrow YZ$

$X \rightarrow XZ(Augementation : X \rightarrow Z); XZ \rightarrow YZ(Augementation : X \rightarrow Y);$
$X \rightarrow YZ(Transitivity : X \rightarrow XZ; XZ \rightarrow YZ)$

# Question 1:

Consider a relational schema $\underline{A}BCDEFGHIJ$, which contains the following FDs:

$$AB \rightarrow C, D \rightarrow E, AE \rightarrow G, GD \rightarrow H, IF \rightarrow J$$

(Don't forget: $A \rightarrow ABCDEFGHIJ$)

Task 2: Dependency preserving decomposition in 3NF

$\underline{A}BCDFGI \quad \underline{D}E \quad \underline{IF}J \quad \underline{GD}H$

The decomposition is dependency preserving, because all FDs in the original set of functional dependencies are implied, if we enforce functional dependencies on the tables $\underline{A}BCDFGI \quad \underline{D}E$ $\underline{IF}J \quad \underline{GD}H$ separately.

For example, $A \rightarrow H$ is ensured, since we can enforce $A \rightarrow GD$ on table $\underline{A}BCDFGI$ and $GD \rightarrow H$ on table $\underline{GD}H$.

# Question 1:

Consider a relational schema $\underline{A}BCDEFGHIJ$, which contains the following FDs:

$$AB \rightarrow C, D \rightarrow E, AE \rightarrow G, GD \rightarrow H, IF \rightarrow J$$

Task 3: This dependency preserving decomposition in 3NF are also in BCNF !!! You can check for each table in this decomposition if it is in BCNF using following definition:

## Definition

R is in Boyce-Codd Normal form if, for every FD $X \rightarrow A$ in F, one of the following statements is true:

- $A \in X$; that is, it is a trivial FD, or
- X is a key or superkey.

# Question 1:

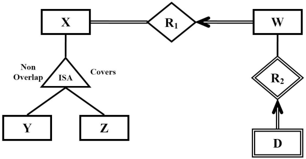



Figure: ER Model

- In the ER diagram above, the diamond labelled $R_1$ denotes:
  - c) A Relationship

- Based on the ER diagram, find the correct description
  - a) For each entity $x \in X$, there exists at least one entity $w \in W$, such that $xR_1w$

# Question 2:

What do the ACID properties stand for?

- Atomicity: All actions in the transaction happen, or none
- Consistency: If each transaction is consistent, and the DB starts consistent, it ends up consistent
- Isolation: Execution of one transaction is isolated from that of all others
- Durability: If a transaction commits, its effects persist

# Question 3:

What are the serial schedule, equivalent schedules and serializable schedule?

Check the book or slides.

# Question 4:

How many file organizations exist? If possible, please point out their advantages and disadvantages?

Check the book or slides.

# Question 5:

What are/is the primary goal(s) in schema normalization?

c) Avoid redundancy

# Question 6:

How does a write ahead log (WAL) work?

b) It stores adds an entry before each write operation to the log-file!

Note, that after commit and abort of a transaction, also information is written to the log file.

# Question 7:

Three-tier architecture

- Presentation Tier: Users require a natural interface to make requests, provide input, and to see results. JavaScript.
- Middle Tier: The application logic executes here. Servlets, JSP, XSLT
- Data Management Tier: Data-intensive Web application involve DBMSs. XML

# Question 8:

Describe briefly about JDBC program

Check JDBC slide !!!

# Good luck for your final DB Exam :)

- Any questions about "Sample Exam Solution", please contact us asap.
- Examination Date: 19 June, 2007 (10:00 $\sim$ 13:00)
- Location: Liacs, Room 174/402