Lecture E1: Review

CS 1555: Database Management Systems

Constantinos Costa
http://db.cs.pitt.edu/courses/cs1555/current.term/

Feb 28, 2019, 16:00-17:15
University of Pittsburgh, Pittsburgh, PA

Lectures based: P. Chrysanthis & N. Farnan Lectures

ACID

- Atomicity
  - Either all the operations associated with a transaction happen or none of them happens
- Consistency Preservation
  - A transaction is a correct program segment. It satisfies the database's integrity constraints at its boundaries
- Isolation
  - Transactions are independent, the result of the execution of concurrent transactions is the same as if transactions were executed serially, one after the other
- Durability (a.k.a. Permanency)
  - The effects of completed transactions become permanent surviving any subsequent failure(s)

Useful Terms

- Cardinality of a relation r(R): # of tuples in r(R) (denoted by |r(R)|)
- Arity or degree of r(R): # of attributes in R (denoted by |R|)

|r(R)| = 3

|R| = 4

SID	*Degree*	*Major*	*Year*
123	BS	Math	1992
064	BA	History	1991
445	PhD	CS	1999

- **|r(R)| ≥ 0 And |R| > 0**
- **Cardinality is property of a relation**
- **Arity is property of relation schema or a relation**

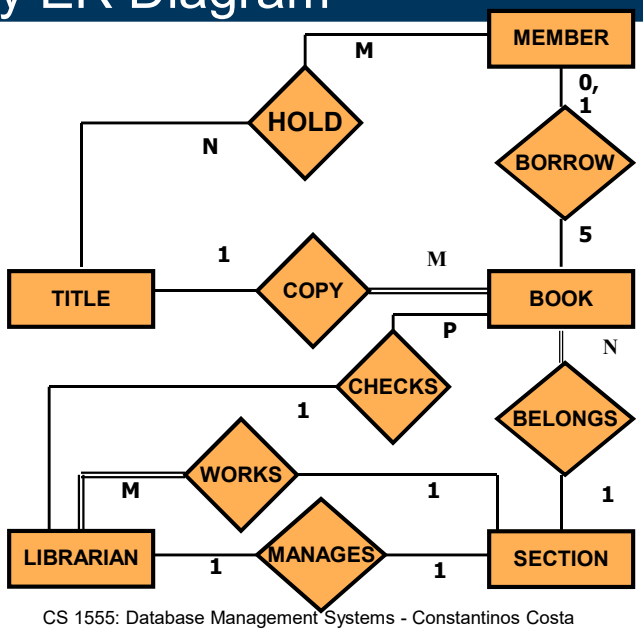
Relational Database Schema

- **A *database schema* is a set of relation schemas**

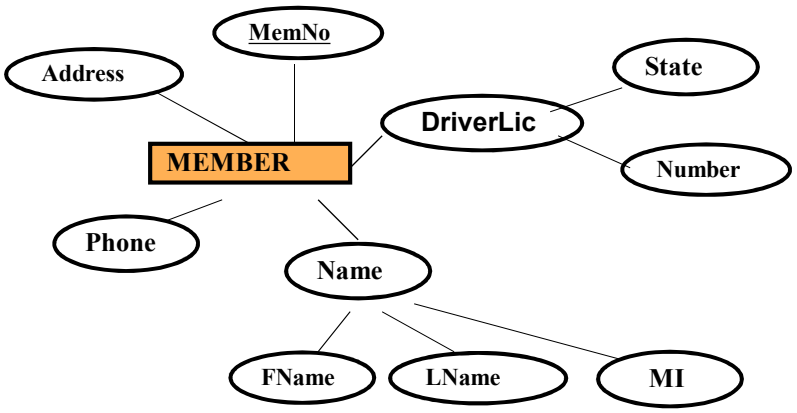
and a set of *integrity constraints*

- Integrity Constraints
  - *Structural* Integrity Constraints
    - key constraints: uniqueness of keys
    - entity integrity constraint: no primary key value can be **NULL**
    - referential integrity constraint
  - *Semantic* Integrity Constraints
    - E.g., ??

Library ER Diagram



Library ER Diagram…



Functional dependencies

- How do we split a bad table?
- How do we recognize a bad table?
- Set of rules: *normal forms*
- **Functional Dependencies** (FD)
- Let R=(A_1, A_2,…, A_n) and X⊆R and Y⊆R
  X → Y if the value of X **uniquely** determines a value of Y (If **t1[X]=t2[X]**, then **t1[Y]=t2[Y])**
  - X functionally determines Y
  - Y is functionally dependent on X

Normal Forms

- 1NF: First Normal Form
  Every attribute has a single atomic value.
- 2NF: Second Normal Form
  It is in 1NF and does not have partial dependencies.
  - Counter Example: 1NF but not 2NF
    SUPPLY (SID, PID, DID, SCity, DCity, Qty)
- 3NF: Third Normal Form
  It is in 2NF and does not have transitive dependencies to attributes that are not part of a key.
  - If X→A is an FD then (a) it is trivial, or (b) X is a superKey, or (c) A is a subset of a **candidate** Key.
  - Counter Example: 2NF but not 3NF
    EMP (SSN, EName, Bdate, Salary, DID, DName, MGRSSN)

Relational Algebra

- Operations on entire relations
  - Operands are (constant or variable) relations
  - Result is a relation
- Set theory operations:
  - Union, Intersection, Difference and Cartesian Product (product for short)
- Specific relational operations:
  - Selection, Projection, Join and Division
- Complete set of relational algebra operations:
  - Select, project, product, union and difference
- SQL is based on concepts from relational algebra

Relational Algebra

- Suppose relation Student has 20 tuples. What is the minimum and maximum number of tuples in the result of this expression:

$$\rho_{s1(i1,n1,g,h)} Student \bowtie \rho_{s2(i2,n2,g,h)} Student$$

a) minimum = 0, maximum = 400
b) minimum = 20, maximum = 20
c) minimum = 20, maximum = 400 ✔
d) minimum = 40, maximum = 40

Relational Algebra

- Which of the following English sentences describes the result of this expression:

$$\pi_{cName} College - \pi_{cName}(Apply \bowtie (\pi_{sID}(\sigma_{GPA>3.5} Student) \cap \pi_{sID}(\sigma_{major='CS'} Apply)))$$

a) All colleges with no GPA>3.5 applicants who applied for a CS major at that college
b) All colleges with no GPA>3.5 applicants who applied for a CS major at any college ✔
c) All colleges where all applicants either have GPA>3.5 or applied for a CS major at that college
d) All colleges where no applicants have GPA>3.5 or no applicants applied for a CS major at that college

Outer Join Examples

- STUDENT(SID, Name, Class, Major)
  ENROLLS(CID, SID, Term, Grade)
- Q1:

```
SELECT *
FROM   (STUDENT S LEFT OUTER JOIN ENROLLS E
                                  ON S.SID=E.SID)
ORDER BY S.SID;
```

- Q2:

```
SELECT SID, S.Name, S. Major
FROM   STUDENT S NATURAL LEFT OUTER JOIN ENROLLS E
WHERE  E.Term IS NULL;
```

Outer Join Q1 Execution

Students

SID	Name	Class	Major
123	John	3	CS
124	Mary	3	CS
999	Newman	1	CS

Enroll

SID	CID	Term	Grade
123	CS1520	Fall 10	3.75
124	CS1520	Fall 10	4
123	CS1555	Fall 10	4
124	CS1555	Fall 10	NULL

Q1 RESULT

S.SID	S.Name	S.Class	S. Major	E.SID	E.CID	E.Term	E.Grade
123	John	3	CS	123	CS1520	Fall 10	3.75
123	John	3	CS	123	CS1555	Fall 10	4
124	Mary	3	CS	124	CS1520	Fall 10	4
124	Mary	3	CS	124	CS1555	Fall 10	NULL
999	Newman	1	CS	NULL	NULL	NULL	NULL

Outer Join Q2 Execution

- **SELECT** SID, S.Name, S. Major
  FROM STUDENT S NATURAL LEFT OUTER JOIN ENROLLS E
  WHERE E.Term IS NULL;

Students

SID	Name	Class	Major
123	John	3	CS
124	Mary	3	CS
999	Newman	1	CS

Enroll

SID	CID	Term	Grade
123	CS1520	Fall 10	3.75
124	CS1520	Fall 10	4
123	CS1555	Fall 10	4
124	CS1555	Fall 10	NULL

Q2 RESULT

S.SID	S.Name	S. Major
999	Newman	CS

Pattern Matching…

- Retrieve all students with *local* phone numbers (any area code) which start with 6 and whose third digit is 3.

```
SELECT  Name
FROM    STUDENT
WHERE   Phone LIKE '____6_3%';
```

- Escape defines the escape character that causes SQL to interpret a wildcard char (%) as itself in a string:

```
SELECT  VideoName
FROM    RENTALS
WHERE   Discount LIKE E'10&%';
```

Natural JOIN

- Suppose relation R(A,C) has the following tuples and relation S(B,C,D) has the following tuples:
- Compute the natural join of R and S. Which of the following tuples is in the result? Assume each tuple has schema (A,B,C,D).

A	C
3	3
6	4
2	3
3	5
7	1

B	C	D
5	1	6
1	5	8
4	3	9

a) (6, 4, 3, 9)
b) (2, 4, 3, 9) ✔
c) (2, 3, 1, 6)
d) (5, 1, 6, 4)

Theta JOIN

- Suppose relation R(A,C) has the following tuples and relation S(B,C,D) has the following tuples:
- Compute the theta-join of R and S with the condition R.B = S.B AND R.A < S.C Which of the following tuples is in the result? Assume each tuple has schema (A, R.B, S.B, C, D).

A	B
1	a
7	t
2	g
4	c
9	t

B	C	D
c	5	6
a	7	8
t	8	9

a) (2, g, c, 5, 6)
b) (4, c, c, 7, 8)
c) (1, a, c, 5, 6)
d) (4, c, c, 5, 6) ✔

Projection

- Suppose relation R(A,B,C) has the following tuples
- Compute the projection π C,B (R). Which of the following tuples is in the result?

A	B	C
1	2	3
4	2	3
4	5	6
2	5	3
1	2	6

a) (6,2) ✔
b) (5,6)
c) (5,3)
d) (2,6)

Difference

- Suppose relation R(A,B,C) has the following tuples and relation S(A,B,C) has the following tuples:
- Compute (R - S) union (S - R), often called the "symmetric difference" of R and S. Which of the following tuples is in the result?

A	B	C
1	2	3
4	2	3
4	5	6
2	5	3
1	2	6

A	B	C
2	5	3
2	5	4
4	5	6
1	2	3

a) (1,2,3)
b) (2,5,4) ✔
c) (4,5,6)
d) (1,5,6)

Intersection

- Suppose relation R(A,B,C) has the following tuples and relation S(A,B,C) has the following tuples:
- Compute the intersection of the relations R and S. Which of the following tuples is in the result?

A	B	C
1	2	3
4	2	3
4	5	6
2	5	3
1	2	6

A	B	C
2	5	3
2	5	4
4	5	6
1	2	3

a) (2,2,6)
b) (2,5,4)
c) (1,2,3) ✔
d) (2,4,3)

Transactions

T_1: `UPDATE` Accounts `SET` balance= balance + 100
`WHERE` client=7

T_2: `UPDATE` Accounts `SET` balance= balance + 500
`WHERE` client=7

- Assume that initially, balance = $1000
- What is the balance after executing T_1 & T_2?
  - should be $1600

However things might go wrong!!

Interleaved Transactions

T_1: `UPDATE` Accounts `SET` balance= balance + 100 `WHERE` client=7

T_2: `UPDATE` Accounts `SET` balance= balance + 500 `WHERE` client=7

- Update (balance) =

  Read (balance); Modify (balance); Write (balance)
- Again, assume that initially balance = $1000
- What happens if T_1 and T_2 are executed concurrently and they both issue Read (balance) at the same time?
  - If T_1 finishes last; balance = $1100
  - If T_2 finishes last, balance = $1500
  - And both values are incorrect!

Query Rewriting

View

```
CREATE VIEW CS_Students AS
SELECT name, age
FROM Student
WHERE Major = 'CS' ;
```

Original Query (user)

```
SELECT name
FROM CS_Students
where age > 19;
```

Modified Query (DBMS)

```
SELECT name
FROM Student
WHERE Major = 'CS'
AND  age>19;
```

Views

- Consider tables R(A,B) and S(B,C) and a view V = select A,C from R,S where R.B=S.B. Suppose R={(1,5),(2,5)} and S={(5,10)}, so V={(1,10),(2,10)}. The user wants to delete tuple (2,10) from V. Which of the following modifications to R and/or S does NOT correctly reflect this modification?

a) delete (2,5) from R

b) update (2,5) to (2,6) in R

c) update (2,5) to (1,6) in R

d) delete (5,10) from S

Views

- Consider tables R(A,B) and S(B,C) and a query Q = select A,C from R,S where R.B=S.B and A < 10 and C > 20. Which of the following materialzed views can NOT be used to help evaluate Q?

a) V1 = select A,C from R,S where R.B=S.B
b) V2 = select A,C from R,S where A < 10 and C > 20 ✔
c) V3 = select A,R.B,S.B,C from R,S where A < 10 and C > 20
d) V4 = select * from R where A < 10

Access control examples

- DBA:
  - GRANT SELECT,INSERT ON Students TO Alice WITH GRANT OPTION;
  - CREATE ROLE Readers;
  - GRANT Readers TO Bob;
  - GRANT Readers TO Charlie;
  - GRANT SELECT ON Students TO Readers;
- Alice:
  - GRANT SELECT,INSERT ON Students TO Bob, Charlie;
- DBA:
  - REVOKE Readers FROM Bob;
  - REVOKE Readers FROM Charlie;
  - GRANT SELECT ON Students TO Charlie;
  - REVOKE ALL PRIVILEGES ON Students FROM ALICE CASCADE;

Chicken and Egg problem

```
CREATE TABLE Chicken (ID INT PRIMARY KEY, eID INT);
CREATE TABLE Egg(ID INT PRIMARY KEY, cID INT);
ALTER TABLE Chicken ADD CONSTRAINT Chicken_FK
            FOREIGN KEY (eID) REFERENCES Egg(ID)
            DEFERRABLE INITIALLY IMMEDIATE;
ALTER TABLE Egg ADD CONSTRAINT Egg_FK
            FOREIGN KEY (cID) REFERENCES Chicken(ID)
            DEFERRABLE INITIALLY IMMEDIATE;
```