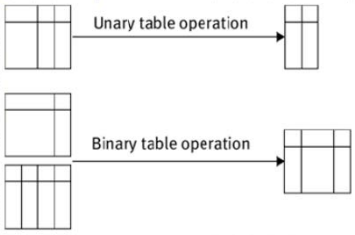
Database Systems

SQL: Data Manipulation (DML)
SQL Set Operations

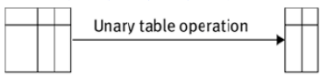
Relational Algebra

- Relational algebraic operations:
  - Projection
  - Restriction
  - Product
  - Join
  - Division
  - Union
  - Intersection
  - Difference



Relational Algebra

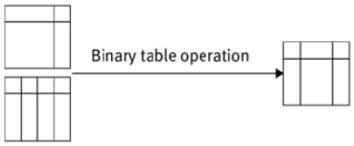
- Unary operators:



  - Projection – pick out columns
  - SQL: SELECT <columns>
  - Restriction – pick out rows meeting certain conditions
  - SQL: WHERE <conditions>

Relational Algebra

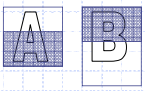
- Binary operators



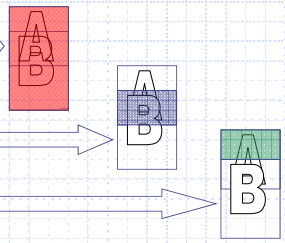
  - Product – SQL: CROSS JOIN
  - Join – SQL: INNER JOIN, OUTER JOIN
    SQL: FROM clause
  - Division – no single SQL counterpart (later)

Relational Algebra

- Binary SET operators
  Union Compatible



  - A Union B :
    All rows from both tables
  - A Intersect B :
    Rows existed in both tables
  - A Difference B :
    Rows in one but not in other table



Union, Intersect, and Difference

- SQL2 defines set operations to combine results of two or more queries into a single result table:
  - Union of two tables, A and B, is table containing all rows in either A or B or both
  - Intersection is table containing all rows common to both A and B
  - Difference (Except) is table containing all rows in A but not in B
- Two tables must be union compatible
- SQL Server 2000 supports only UNION [ALL]

Union, Intersect, and Difference

- Format of set operator clause in each case is:

```
subquery1
setOp [ALL] [CORRESPONDING
            [BY {column1 [, ...]}]]
subquery2
```

- If ALL specified, result include duplicate rows
- If CORRESPONDING BY specified, set operation performed on the named column(s)
- If CORRESPONDING specified but not BY clause, operation performed on common columns

Books Database Schema

authors (au_id, au_fname, au_lname, phone, address, city, state, zip)

title_authors (title_id, au_id, au_order, royalty_share)

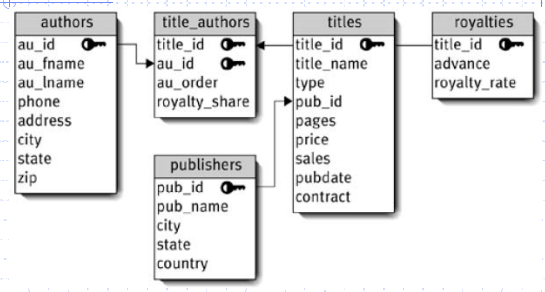
publishers (pub_id, pub_name, city, state, country)

royalties (title_id,advance,royalty_rate)

titles (title_id, title_name, type, pub_id, pages, price, sales, pubdate, contract)

http://www.fehily.com/books/sql2/downloads.htm

Books Database Schema



UNION

- List all cities where there is either an author or a publisher:

```
SELECT city FROM authors
  WHERE city IS NOT NULL
UNION
SELECT city FROM publishers
  WHERE city IS NOT NULL;
```

- OR

```
SELECT city FROM authors
  WHERE city IS NOT NULL
UNION ALL
SELECT city FROM publishers
  WHERE city IS NOT NULL;
```


INTERSECT

- List all cities where there is both an author and a publisher:

```
SELECT city FROM authors
INTERSECT
SELECT city FROM publishers;
```

- Or

```
SELECT DISTINCT city
FROM authors
WHERE city IN (SELECT city
               FROM publishers);
```


INTERSECT – SQL1

- Could rewrite this query without INTERSECT:

```
SELECT DISTINCT a.city
FROM authors a
  JOIN publishers p ON a.city = p.city;
```

- Or:

```
SELECT DISTINCT city
FROM authors a
WHERE EXISTS
  (SELECT *
   FROM publishers p
   WHERE a.city = p.city);
```

INTERSECT – Alternatives

- Using JOIN
- Using Subquery with IN
- Using Subquery with EXISTS
- Using INTERSECT if supported

Difference : EXCEPT / MINUS

- List of all cities where there is an author but no publisher:

```
SELECT city FROM authors
EXCEPT      -- Oracle uses MINUS
SELECT city FROM publishers;
```

- Or

```
SELECT DISTINCT city
FROM authors
WHERE city NOT IN
  (SELECT city
   FROM publishers);
```


EXCEPT - SQL1

- Could rewrite this query without EXCEPT:

```
SELECT DISTINCT a.city
FROM authors a
     LEFT JOIN publishers p
          ON a.city = p.city
WHERE p.city IS NULL;
```

- Or

```
SELECT DISTINCT city
FROM authors a
WHERE NOT EXISTS
  (SELECT *
   FROM publishers p
   WHERE a.city = p.city);
```

EXCEPT - Alternatives

- Using OUTER JOIN
- Using Subquery with NOT IN
- Using Subquery with NOT EXISTS
- Using EXCEPT or MINUS if supported

Another Example

```
SQL> SELECT * FROM mypubs;

PUB PUB_NAME
--- --------------------
P01 Pearson
P02 McGraw-Hill
P03 Morgan-Kaufmann
P04 New World Book

SQL> SELECT * FROM mytitles;

TIT TITLE_NAME   TYPE       PUB PUBDATE
--- ------------ ---------- --- ----------
T01 1977!        history    P01 2000-08-01
T02 World War I  history    P03 1998-04-01
T03 Database     computer   P03 2000-09-01
T05 Your Ideas   psychology P01 2001-01-01
T10 Mr President biography
T13 Civil War    history    P02 1999-05-31
```

- Find the publishers who have not yet published a book

Using LEFT [OUTER] JOIN

- Find the publishers who have not yet published a book
-
```
SELECT *
FROM mypubs p
   LEFT JOIN mytitles t   -- LEFT JOIN
   ON p.pub_id = t.pub_id
WHERE t.pub_id IS NULL;
```

```
PUB PUB_NAME          TIT TITLE_NAME   TYPE       PUB PUBDATE
--- ----------------- --- ------------ ---------- --- ----------
P01 Pearson           T01 1977!        history    P01 2000-08-01
P03 Morgan-Kaufmann   T02 World War I  history    P03 1998-04-01
P03 Morgan-Kaufmann   T03 Database     computer   P03 2000-09-01
P01 Pearson           T05 Your Ideas   psychology P01 2001-01-01
P02 McGraw-Hill       T13 Civil War    history    P02 1999-05-31
P04 New World Book
```

Using EXCEPT / MINUS

- Find the publishers who have not yet published a book
-
```
SELECT pub_id, pub_name
FROM mypubs

MINUS

SELECT p.pub_id, pub_name
FROM mypubs p
     JOIN mytitles t
       ON p.pub_id =
          t.pub_id
;
```

```
PUB PUB_NAME
--- --------------------
P01 Pearson
P02 McGraw-Hill
P03 Morgan-Kaufmann
P04 New World Book
```

MINUS

```
PUB PUB_NAME
--- --------------------
P01 Pearson
P01 Pearson
P02 McGraw-Hill
P03 Morgan-Kaufmann
P03 Morgan-Kaufmann
```


Using NOT IN

- Find the publishers who have not yet published a book
-
```
SELECT pub_id, pub_name
FROM mypubs

WHERE pub_id NOT IN

   (SELECT pub_id
    FROM mytitles
   )
;
```

```
PUB PUB_NAME
--- --------------------
P01 Pearson
P02 McGraw-Hill
P03 Morgan-Kaufmann
P04 New World Book
```

NOT IN

```
PUB
---
P01
P01
P02
P03
P03
```


Using NOT EXISTS

- Find the publishers who have not yet published a book
-
```
SELECT pub_id, pub_name
FROM mypubs p

WHERE pub_id NOT EXISTS

   (SELECT 1
    FROM mytitles t
    WHERE p.pub_id =
          t.pub_id
   )
;
```

```
PUB PUB_NAME
--- --------------------
P01 Pearson
P02 McGraw-Hill
P03 Morgan-Kaufmann
P04 New World Book
```

NOT EXISTS

```
P01
---   P02
1     ---  P03
1     1    ---  P04
                1    ---
                1
```

VIEW – A Quick Introduction

```
DROP VIEW myview;
CREATE OR REPLACE VIEW myview
AS
  subquery
;
```

```
SELECT *
FROM myview
WHERE <condition>
;
```