Section 2
SQL Queries
Agenda

- Insert, Update, and Delete
- Constraints
- Group by: Group practice
- Sub queries: Discussion
Schema for today’s queries

- AUTHOR(aid, name, age)
- WRITTEN_BY(bid, aid)
- BOOK (bid, title)
- BOOK_WORDS (bid, word)
- WORD(word) - we can do without this.
Insert, Update, and Delete

- **INSERT INTO AUTHOR VALUES**
  (312, ‘Michael Chabon’, 45);

- **UPDATE AUTHOR SET AGE=46 WHERE**
  aid=312;

- **DELETE FROM AUTHOR WHERE**
  aid=312;

Note: for **DELETE** [be careful! don’t forget the WHERE condition!]

Constraints

- What are examples of ICs constraints that we might want?
  - Keys, foreign keys
  - Attribute-level constraints
  - Tuple-level
  - Global constraints

- Policies
  - Reject
  - Cascade
  - Set NULL
Group Exercises 1

- Find names of authors who wrote more than 20 books.

  **Without group-by**
  ```sql
  SELECT name
  FROM AUTHOR a
  WHERE (SELECT COUNT(*)
          FROM WRITTEN_BY wb
          WHERE wb.aid = a.aid) > 20
  ```

  **With group-by:**
  ```sql
  SELECT name
  FROM AUTHOR a, WRITTEN_BY wb
  WHERE a.aid = wb.aid
  GROUP BY a.aid, a.name --note that we must include a.name
  HAVING COUNT(*) > 20
  ```
Group by error

- Column '___' is invalid in the select list because it is not contained in either an aggregate function or the GROUP BY clause.
Group Exercise 2

- Find authors who have a vocabulary of more than 10k words.

**Without group-by:**

```sql
SELECT name
FROM AUTHOR A
WHERE (SELECT COUNT(DISTINCT word) FROM WRITTEN_BY wb, Book_Words bw WHERE A.aid = wb.aid AND wb.bid = bw.bid) > 10000
```

**With group-by:**

```sql
SELECT name
FROM AUTHOR a, WRITTEN_BY wb, BOOK_WORDS bw
WHERE a.aid = wb.aid AND wb.bid = bw.bid
GROUP BY a.aid, a.name
HAVING COUNT(DISTINCT word) > 10000
```
Group Exercise 3

- For each author, report average number of words per book.

- With group-by:

```sql
SELECT aid, AVG(num)
FROM (SELECT aid, bid, COUNT(*) num
    FROM AUTHOR a, WRITTEN_BY wb, BOOK_WORDS bw
    WHERE ...
    GROUP BY aid, bid) t
GROUP BY aid
```
Group Exercise 4

- Find most frequently used word.

- With group-by:

```sql
SELECT  word
FROM    BOOK_WORDS
GROUP BY word
HAVING  count(*) >= ALL  (SELECT count(*)
FROM    BOOK_WORDS bw1
GROUP BY bw1.word)
```
Subqueries: Discussion

- Where can sub-queries occur?
  - SELECT / FROM / WHERE / HAVING

- If I make a subquery S in the _____ (one of the above) clause,
  - Where can I access S?
  - What properties must S have?