

# Midterm Topics

- SQL
- Aggregates
- Nested queries
- Indexes
- Query implementation
- One-pass algorithms
- Cost accounting
- Relational algebra (functions, trees)

# SQL Reminders

```
CREATE TABLE Class (  
    dept VARCHAR(6),  
    number INTEGER,  
    title VARCHAR(75),  
    PRIMARY KEY (dept, number)  
);
```

- SELECT ... FROM ... WHERE ... GROUP BY ...  
 HAVING ...;
- LEFT OUTER JOIN
- WHERE name LIKE '%e%'
- ORDER BY attribute1 DESC, attribute2 ASC

# SQL Aggregates

```
SELECT column1, aggregate(column2)
FROM table
GROUP BY column1;
```

- COUNT(...)
- SUM(...)
- MIN(...)
- MAX(...)

Selected attributes must be in the group by clause or an aggregate function.

Use DISTINCT at the beginning of attribute lists to eliminate duplicates.

# SQL Aggregates

```
SELECT column1, aggregate(column2)
FROM table
WHERE [condition for individual tuples]
GROUP BY column1
HAVING [condition for entire group];
```

# Nested Queries

## **Subqueries in WHERE:**

- SELECT ..... WHERE EXISTS (subquery);
- SELECT ..... WHERE NOT EXISTS (subquery);
- SELECT ..... WHERE attribute IN (subquery);
- SELECT ..... WHERE attribute NOT IN (subquery);
- SELECT ..... WHERE attribute > ANY (subquery);
- SELECT ..... WHERE attribute > ALL (subquery);

# Nested Queries

## **Subqueries in FROM:**

```
SELECT x.pname  
FROM (SELECT * FROM Product WHERE price < 20) AS x  
WHERE x.price < 500;
```

## **Subqueries in SELECT:**

```
SELECT DISTINCT C.cname, (SELECT COUNT(*)  
                           FROM Product P  
                           WHERE P.cid = C.cid)  
FROM Company C;
```

# Query Implementation

## Cost Parameters:

$B(R)$  - # of blocks for relation

$T(R)$  - # of tuples

$V(R, a)$  - # of distinct values of attribute  $a$

$M$  - # of memory pages

## One-Pass Algorithms:

- Nested Loop Join
- Hash Join
- Table in Sec. 15.3.5, page 722

## Index-Based Algorithms:

- With or without index
- With or without clustering
- Example 15.11, page 741

# Relational Algebra

Selection:  $\sigma$

Projection:  $\pi$

Join:  $\bowtie$

Group By:  $\gamma$

Set Union:  $\cup$

Set Intersection:  $\cap$

Set Difference:  $-$

Duplicate Elimination:  $\delta$

Renaming:  $\rho$

Sorting:  $\tau$

Be able to translate SQL  $\rightarrow$  RA and RA  $\rightarrow$  SQL.

Be able to write RA as a function and draw it as a tree.