CIS 451/551

week 2: Intro SQL
Beginning SQL

• Basic format
  SELECT <attribute list>
  FROM <table list>
  WHERE <logical condition>

• order of “execution”
  FROM → WHERE → SELECT
example 1.1

SELECT fname, lname, dno
FROM employee
WHERE dno=5 AND fname LIKE 'J%'
example 1.2

SELECT CONCAT(fname, " ", lname) AS fullName
FROM employee
WHERE dno=5 AND fname LIKE 'J%'

CONCAT is a built-in string function
example 1.3

SELECT dname, CONCAT(fname, " ", lname) AS fullName
FROM employee, department
WHERE dno=dnumber
ORDER BY dname

- an example JOIN
- FROM on multiple tables creates a cross product
- employee x department
example 1.4

```sql
SELECT CONCAT(fname, " ", lname) AS fullName, order_date
FROM customer, orders
WHERE customer.customer_num=orders.customer_num
```

another JOIN, note the disambiguation
example 1.5

SELECT CONCAT(c.fname, " ", c.lname) AS fullName,
       MONTHNAME(o.order_date)
FROM customer c, orders o
WHERE c.customer_num = o.customer_num

- MONTHNAME is a date type function
- note aliasing of table names
- can also use AS
example 1.6

SELECT CONCAT(c.fname, " ", c.lname) AS fullName,
        o.order_num, i.item_num, i.manu_code
FROM customer c JOIN orders o
    ON c.customer_num = o.customer_num
JOIN items i ON o.order_num = i.order_num

- three-way JOIN
- JOIN keyword used
- explicitly identifies JOIN to optimizer
references

  (top level SELECT syntax)

  (JOIN syntax)

  (string operators)
join style handout

• look in handouts directory of class web site
• FROM t1, t2, t3 WHERE conditions
• FROM t1 JOIN t2 ON condition1 JOIN t3 ON condition2 (preferred)
• FROM t1 JOIN t2 USING(attributes) JOIN t3 USING(attributes) (preferred also)
• FROM t1 NATURAL JOIN t2 NATURAL JOIN t3

• also INNER JOIN, OUTER JOIN
WARNING: do not forget the join conditions

SELECT attributes
FROM table1 JOIN table2 ON condition1
    JOIN table3 ON condition2
    JOIN table4 ON condition3

here 4 tables and 3 join conditions
example 2.1

```
SELECT  s.stock_num, s.manu_code, s.description
FROM    stock s INNER JOIN items i
        USING(stock_num, manu_code)
```

On stores7, find everything from stock table that has been ordered (it should appear in the items table to have been ordered). Note use of two item key in the USING clause.
SELECT  s.stock_num, s.manu_code, s.description, i.order_num
FROM stock s LEFT JOIN items i
    USING(stock_num, manu_code)

Same query as previous but using LEFT OUTER join. Includes all stock entries that do not meet the join condition (with NULL in fields from items)
example 2.3

```
SELECT  s.stock_num, s.manu_code, s.description, i.order_num
FROM stock s LEFT JOIN items i
    USING(stock_num, manu_code)
WHERE i.order_num IS NULL
```

As previous, but only the entries with NULL fields (so stock items never ordered).

Compare to IS NOT NULL.
example 2.4 (aggregate functions)

SELECT COUNT(*)
FROM stock s LEFT JOIN items i
    USING(stock_num, manu_code)
WHERE i.order_num IS NULL

COUNT(*) counts the number of rows of output, so here number of stock items never ordered.

example 3.1

```sql
SELECT c.fname, c.lname, o.ship_weight
FROM customer c LEFT JOIN orders o USING(customer_num)
```

sample LEFT OUTER join, includes customers not making an order
example 3.2

```
SELECT c.fname, c.lname, o.order_num, i.manu_code
FROM customer c LEFT JOIN
    (orders o JOIN items i USING(order_num) )
USING(customer_num)
```

a way to re-parenthesize so NULL customers don’t get lost
example 3.3

SELECT e1.fname, e1.lname, e2.fname, e2.lname
FROM employee e1 JOIN employee e2
    ON e1.ssn=e2.superssn

example self-join on employee table, match employees with their supervisor (company database)
SELECT c.fname, c.lname, i.manu_code
FROM customer c JOIN orders o USING(customer_num)
    JOIN items i USING(order_num)
WHERE i.manu_code IN ('SHM', 'ANZ')
example 3.5

SELECT c.fname, c.lname, i.manu_code
FROM customer c JOIN orders o USING(customer_num)
    JOIN items i USING(order_num)
WHERE i.manu_code LIKE 'SHM'
UNION
SELECT c.fname, c.lname, i.manu_code
FROM customer c JOIN orders o USING(customer_num)
    JOIN items i USING(order_num)
WHERE i.manu_code LIKE 'ANZ'

• rewriting above with UNION
• UNION removes duplicates
• UNION ALL retains duplicate entries