Lecture 6:
• SQL Query Basics

Announcements:
• HW-1 due Thur
• PS-2 due Tues, Sept 27
• Q-2 on Tues, Sept 27

SQL Query Basics

SQL queries use the SELECT command ... or WITH (more later)
• each query $Q$ is defined over one or more tables
• the result of a query $Q$ is an "output" table (duplicate rows allowed)
• thus, queries can be viewed as functions $Q : R_1 \times \cdots \times R_n \rightarrow R_{output}$

Our Plan:
• cover basic "Select-From-Where" queries this week (plus ordering)
• come back to more advanced query constructs later (e.g., for analytics)
Basic Select-From-Where Queries

### Loan

<table>
<thead>
<tr>
<th>acct_id</th>
<th>barcode</th>
<th>checkout_date</th>
<th>due_date</th>
<th>return_date</th>
</tr>
</thead>
<tbody>
<tr>
<td>101</td>
<td>4242</td>
<td>8/12</td>
<td>8/26</td>
<td>8/24</td>
</tr>
<tr>
<td>101</td>
<td>4243</td>
<td>8/12</td>
<td>8/19</td>
<td>NULL</td>
</tr>
<tr>
<td>102</td>
<td>4242</td>
<td>8/25</td>
<td>9/7</td>
<td>8/29</td>
</tr>
<tr>
<td>101</td>
<td>4243</td>
<td>7/10</td>
<td>7/17</td>
<td>7/18</td>
</tr>
</tbody>
</table>

**Example:** Find barcodes of books loaned by account 101

```
SELECT barcode
FROM Loan
WHERE acct_id = 101
```

*Note:* three barcodes returned in a 1-column table!

SQL "Select-From-Where" queries

1. **SELECT** specifies attribute(s) to return
2. **FROM** lists table(s) being queried ... can think "row-by-row"
3. **WHERE** places (Boolean) conditions on rows to return

WHERE conditions

**Example:** Find due dates of copies of book 4243 checked out by account 101

```
SELECT due_date
FROM Loan
WHERE acct_id = 101 AND
    barcode = 4243
```

Basic WHERE clause conditions:

- expressions involving **AND, OR, NOT** ... can be parenthesized
- relational comparators: =, <, >, <=, >=, != ... or <> for !=
- also **BETWEEN x AND y** ... e.g., barcode BETWEEN 4200 AND 4300
- **IS NULL** instead of = NULL ... more on NULLs later
- **IS NOT NULL** insted of != NULL
WHERE conditions

Check in: write a query for barcodes loaned to account 101 that are either still checked out or were returned after the due date

```
SELECT barcode
FROM Loan
WHERE acct_id = 101 AND
  (return_date IS NULL OR return_date > due_date)
```

• must use IS NULL here (more later)

SELECT attributes

SELECT can have more than one attribute ...

```
SELECT barcode, due_date
FROM Loan
WHERE acct_id = 101
```

The query returns a result table with two attributes:

<table>
<thead>
<tr>
<th>barcode</th>
<th>due_date</th>
</tr>
</thead>
<tbody>
<tr>
<td>4242</td>
<td>8/26</td>
</tr>
<tr>
<td>4243</td>
<td>8/19</td>
</tr>
<tr>
<td>4243</td>
<td>7/17</td>
</tr>
</tbody>
</table>

A "*" means return all attributes ...

```
SELECT *
FROM Loan
WHERE acct_id = 101
```
SELECT attributes

Result attributes can be renamed ...

SELECT acct_id AS account, barcode AS returned_book
FROM Loan
WHERE return_date IS NOT NULL

Which returns:

<table>
<thead>
<tr>
<th>acctout</th>
<th>returned_book</th>
</tr>
</thead>
<tbody>
<tr>
<td>101</td>
<td>4242</td>
</tr>
<tr>
<td>102</td>
<td>4242</td>
</tr>
<tr>
<td>101</td>
<td>4243</td>
</tr>
</tbody>
</table>

Ordering Results by Attribute

Result can be ordered using an ORDER BY clause ...

SELECT acct_id, barcode, due_date
FROM loan
WHERE acct_id = 101
ORDER BY due_date ASC -- ASC = ascending (default, can leave ASC off)

Which returns:

<table>
<thead>
<tr>
<th>acct_id</th>
<th>barcode</th>
<th>due_date</th>
</tr>
</thead>
<tbody>
<tr>
<td>101</td>
<td>4243</td>
<td>2022-07-17</td>
</tr>
<tr>
<td>102</td>
<td>4243</td>
<td>2022-08-19</td>
</tr>
<tr>
<td>101</td>
<td>4242</td>
<td>2022-08-26</td>
</tr>
</tbody>
</table>

Instead of ascending (smallest-to-largest) can also use DESC (descending):

SELECT acct_id, barcode, due_date
FROM loan
WHERE acct_id = 101
ORDER BY due_date DESC -- DESC = descending
Ordering Results by Attribute (cont)

Can sort on multiple attributes to “break ties” ...

```sql
SELECT acct_id, barcode, due_date
FROM loan
ORDER BY acct_id, due_date DESC -- more than 2 allowed
```

which returns:

<table>
<thead>
<tr>
<th>acct_id</th>
<th>barcode</th>
<th>due_date</th>
</tr>
</thead>
<tbody>
<tr>
<td>101</td>
<td>4242</td>
<td>2022-08-26</td>
</tr>
<tr>
<td>101</td>
<td>4243</td>
<td>2022-08-19</td>
</tr>
<tr>
<td>101</td>
<td>4243</td>
<td>2022-07-17</td>
</tr>
<tr>
<td>102</td>
<td>4242</td>
<td>2022-09-07</td>
</tr>
</tbody>
</table>

Note: Only SELECT and FROM are required in a query

- and in MySQL only SELECT!

... e.g: SELECT 'hello world'