SQL Queries
(Chapter 3.3)

Peter Revesz

CSCE 413/813
Computer Science and Engineering
University of Nebraska – Lincoln
# Hospital Database

## Patient

<table>
<thead>
<tr>
<th>Name</th>
<th>ID</th>
<th>CM</th>
<th>KG</th>
</tr>
</thead>
<tbody>
<tr>
<td>Anderson</td>
<td>100</td>
<td>200</td>
<td>130</td>
</tr>
<tr>
<td>Brown</td>
<td>111</td>
<td>150</td>
<td>50</td>
</tr>
<tr>
<td>Davis</td>
<td>222</td>
<td>190</td>
<td>90</td>
</tr>
<tr>
<td>Edwards</td>
<td>333</td>
<td>160</td>
<td>90</td>
</tr>
<tr>
<td>Ford</td>
<td>345</td>
<td>165</td>
<td>100</td>
</tr>
<tr>
<td>Hardy</td>
<td>454</td>
<td>175</td>
<td>70</td>
</tr>
<tr>
<td>Johnson</td>
<td>567</td>
<td>170</td>
<td>50</td>
</tr>
<tr>
<td>Smith</td>
<td>755</td>
<td>180</td>
<td>120</td>
</tr>
</tbody>
</table>

## Doctor

<table>
<thead>
<tr>
<th>Name</th>
<th>ID</th>
<th>Age</th>
<th>Specialty</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cheney</td>
<td>987</td>
<td>50</td>
<td>pediatrics</td>
</tr>
<tr>
<td>Hardy</td>
<td>454</td>
<td>53</td>
<td>osteopathology</td>
</tr>
<tr>
<td>McBride</td>
<td>377</td>
<td>36</td>
<td>radiology</td>
</tr>
<tr>
<td>Miller</td>
<td>300</td>
<td>60</td>
<td>neurology</td>
</tr>
<tr>
<td>Moss</td>
<td>244</td>
<td>30</td>
<td>neurology</td>
</tr>
<tr>
<td>Nelson</td>
<td>400</td>
<td>76</td>
<td>cardiology</td>
</tr>
<tr>
<td>Oltman</td>
<td>181</td>
<td>56</td>
<td>urology</td>
</tr>
<tr>
<td>Paine</td>
<td>266</td>
<td>45</td>
<td>cardiology</td>
</tr>
<tr>
<td>Pepper</td>
<td>555</td>
<td>42</td>
<td>cardiology</td>
</tr>
<tr>
<td>Snow</td>
<td>500</td>
<td>65</td>
<td>radiology</td>
</tr>
</tbody>
</table>

## Visit

<table>
<thead>
<tr>
<th>PID</th>
<th>DID</th>
<th>Month</th>
<th>Day</th>
<th>Year</th>
</tr>
</thead>
<tbody>
<tr>
<td>100</td>
<td>181</td>
<td>5</td>
<td>20</td>
<td>2008</td>
</tr>
<tr>
<td>100</td>
<td>555</td>
<td>6</td>
<td>30</td>
<td>2009</td>
</tr>
<tr>
<td>111</td>
<td>987</td>
<td>8</td>
<td>20</td>
<td>2009</td>
</tr>
<tr>
<td>111</td>
<td>987</td>
<td>5</td>
<td>28</td>
<td>2010</td>
</tr>
<tr>
<td>222</td>
<td>266</td>
<td>9</td>
<td>12</td>
<td>2007</td>
</tr>
<tr>
<td>222</td>
<td>400</td>
<td>5</td>
<td>20</td>
<td>2008</td>
</tr>
<tr>
<td>222</td>
<td>555</td>
<td>5</td>
<td>20</td>
<td>2008</td>
</tr>
<tr>
<td>333</td>
<td>987</td>
<td>6</td>
<td>23</td>
<td>2009</td>
</tr>
<tr>
<td>345</td>
<td>300</td>
<td>5</td>
<td>16</td>
<td>2009</td>
</tr>
<tr>
<td>454</td>
<td>244</td>
<td>6</td>
<td>10</td>
<td>2010</td>
</tr>
<tr>
<td>567</td>
<td>377</td>
<td>2</td>
<td>20</td>
<td>2010</td>
</tr>
<tr>
<td>567</td>
<td>454</td>
<td>5</td>
<td>28</td>
<td>2010</td>
</tr>
<tr>
<td>755</td>
<td>987</td>
<td>6</td>
<td>23</td>
<td>2009</td>
</tr>
</tbody>
</table>
Basic SQL Queries

Basic:

\[
\text{CREATE VIEW} \quad R(B_1, \ldots, B_m) \\
\text{SELECT} \quad A_1, \ldots, A_n \\
\text{FROM} \quad R_1, \ldots, R_l \\
\text{WHERE} \quad C_1 \text{ AND } \ldots \text{ AND } C_o
\]

The CREATE VIEW and WHERE clauses are optional.

Example: Find the specialty of all the doctors in the hospital.

\[
\text{SELECT} \quad \text{Specialty} \\
\text{FROM} \quad \text{Doctor}
\]

<table>
<thead>
<tr>
<th>Specialty</th>
</tr>
</thead>
<tbody>
<tr>
<td>pediatrics</td>
</tr>
<tr>
<td>osteopathology</td>
</tr>
<tr>
<td>radiology</td>
</tr>
<tr>
<td>neurology</td>
</tr>
<tr>
<td>neurology</td>
</tr>
<tr>
<td>cardiology</td>
</tr>
<tr>
<td>urology</td>
</tr>
<tr>
<td>cardiology</td>
</tr>
<tr>
<td>cardiology</td>
</tr>
<tr>
<td>radiology</td>
</tr>
</tbody>
</table>
Basic SQL Queries

We can use also an optional DISTINCT keyword.

Example: Find the specialty of all the doctors in the hospital.

SELECT DISTINCT Specialty
FROM Doctor

<table>
<thead>
<tr>
<th>Specialty</th>
</tr>
</thead>
<tbody>
<tr>
<td>pediatrics</td>
</tr>
<tr>
<td>osteopathology</td>
</tr>
<tr>
<td>radiology</td>
</tr>
<tr>
<td>neurology</td>
</tr>
<tr>
<td>cardiology</td>
</tr>
<tr>
<td>urology</td>
</tr>
</tbody>
</table>

This is equivalent to a projection in relational algebra.
Basic SQL Queries

Example: Find when each patient visited the hospital.

```
FROM     Patient, Visit
WHERE    Patient.ID = Visit.PID
```

<table>
<thead>
<tr>
<th>Patient.Name</th>
<th>Visit.Month</th>
<th>Visit.Day</th>
<th>Visit.Year</th>
</tr>
</thead>
<tbody>
<tr>
<td>Anderson</td>
<td>5</td>
<td>20</td>
<td>2008</td>
</tr>
<tr>
<td>Anderson</td>
<td>6</td>
<td>30</td>
<td>2009</td>
</tr>
<tr>
<td>Brown</td>
<td>8</td>
<td>20</td>
<td>2009</td>
</tr>
<tr>
<td>Brown</td>
<td>5</td>
<td>28</td>
<td>2010</td>
</tr>
<tr>
<td>Davis</td>
<td>9</td>
<td>12</td>
<td>2007</td>
</tr>
<tr>
<td>Davis</td>
<td>5</td>
<td>20</td>
<td>2008</td>
</tr>
<tr>
<td>Davis</td>
<td>5</td>
<td>20</td>
<td>2008</td>
</tr>
<tr>
<td>Edwards</td>
<td>6</td>
<td>23</td>
<td>2009</td>
</tr>
<tr>
<td>Ford</td>
<td>5</td>
<td>16</td>
<td>2009</td>
</tr>
<tr>
<td>Hardy</td>
<td>6</td>
<td>10</td>
<td>2010</td>
</tr>
<tr>
<td>Johnson</td>
<td>2</td>
<td>20</td>
<td>2010</td>
</tr>
<tr>
<td>Johnson</td>
<td>5</td>
<td>28</td>
<td>2010</td>
</tr>
<tr>
<td>Smith</td>
<td>6</td>
<td>23</td>
<td>2009</td>
</tr>
</tbody>
</table>

This is equivalent to a join followed by a projection in relational algebra.
Basic SQL Queries

Example: Find which doctor treats which patient.

```
CREATE VIEW Treats(DoctorName, PatientName)
SELECT D.Name, P.Name,
FROM Patient AS P, Doctor AS D, Visit AS V
WHERE P.ID = V.PID AND D.ID = V.DID
```

Treats

<table>
<thead>
<tr>
<th>DoctorName</th>
<th>PatientName</th>
</tr>
</thead>
<tbody>
<tr>
<td>Oltman</td>
<td>Anderson</td>
</tr>
<tr>
<td>Pepper</td>
<td>Anderson</td>
</tr>
<tr>
<td>Cheney</td>
<td>Brown</td>
</tr>
<tr>
<td>Cheney</td>
<td>Brown</td>
</tr>
<tr>
<td>Paine</td>
<td>Davis</td>
</tr>
<tr>
<td>Nelson</td>
<td>Davis</td>
</tr>
<tr>
<td>Pepper</td>
<td>Davis</td>
</tr>
<tr>
<td>Cheney</td>
<td>Edwards</td>
</tr>
<tr>
<td>Miller</td>
<td>Ford</td>
</tr>
<tr>
<td>Moss</td>
<td>Hardy</td>
</tr>
<tr>
<td>McBride</td>
<td>Johnson</td>
</tr>
<tr>
<td>Hardy</td>
<td>Johnson</td>
</tr>
<tr>
<td>Cheney</td>
<td>Smith</td>
</tr>
</tbody>
</table>
```
Aggregation SQL Queries

Aggregation Operators take as input a set of values and give as output a single value. Examples: AVG, COUNT, MAX, MIN and SUM.

Example: Find the number of cardiologists.

```
SELECT COUNT(ID)
FROM Doctor
WHERE Specialty = "cardiology"
```

<table>
<thead>
<tr>
<th>Count(ID)</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
</tr>
</tbody>
</table>
The optional GROUP BY clause divides all the records into groups such that in each group the values of the attributes $B_1, B_n$ are the same.

**Example:** Find the number of doctors for each specialty.

```
SELECT Specialty, COUNT(ID)
FROM Doctor
GROUP BY Specialty
```

<table>
<thead>
<tr>
<th>Name</th>
<th>ID</th>
<th>Age</th>
<th>Specialty</th>
</tr>
</thead>
<tbody>
<tr>
<td>Nelson</td>
<td>400</td>
<td>76</td>
<td>cardiology</td>
</tr>
<tr>
<td>Paine</td>
<td>266</td>
<td>45</td>
<td>cardiology</td>
</tr>
<tr>
<td>Pepper</td>
<td>555</td>
<td>42</td>
<td>cardiology</td>
</tr>
<tr>
<td>Miller</td>
<td>300</td>
<td>60</td>
<td>neurology</td>
</tr>
<tr>
<td>Moss</td>
<td>244</td>
<td>30</td>
<td>neurology</td>
</tr>
<tr>
<td>Hardy</td>
<td>454</td>
<td>53</td>
<td>osteopathology</td>
</tr>
<tr>
<td>Cheney</td>
<td>987</td>
<td>50</td>
<td>pediatrics</td>
</tr>
<tr>
<td>McBride</td>
<td>377</td>
<td>36</td>
<td>radiology</td>
</tr>
<tr>
<td>Snow</td>
<td>500</td>
<td>65</td>
<td>radiology</td>
</tr>
<tr>
<td>Oltman</td>
<td>181</td>
<td>56</td>
<td>urology</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Specialty</th>
<th>Count(ID)</th>
</tr>
</thead>
<tbody>
<tr>
<td>cardiology</td>
<td>3</td>
</tr>
<tr>
<td>neurology</td>
<td>2</td>
</tr>
<tr>
<td>osteopathology</td>
<td>1</td>
</tr>
<tr>
<td>pediatrics</td>
<td>1</td>
</tr>
<tr>
<td>radiology</td>
<td>2</td>
</tr>
<tr>
<td>urology</td>
<td>1</td>
</tr>
</tbody>
</table>
SQL Queries with Set Operators

SQL set operator keywords are **UNION**, **INTERSECT** and **MINUS** representing set union, set intersection and set difference.

**Example:** Find the doctors who are also patients.

```sql
SELECT Name
FROM Doctor
INTERSECT
SELECT Name
FROM Patient
```

**Set:**

```
SQL Query 1
SET_OPERATOR
SQL Query 2
```

Name

Hardy
Nested SQL Queries

**Nested:**

```
SELECT ... 
FROM ... 
WHERE ... AND A_i NEST_OPERATOR (SELECT A_j 
FROM ... 
WHERE ...) 
```

**Example:** Find the oldest radiologist in the hospital.

```
SELECT Name 
FROM Doctor 
WHERE Specialty="radiology" 
AND Age >= ALL (SELECT Age 
FROM Doctors 
WHERE Specialty="radiology") 
```

<table>
<thead>
<tr>
<th>Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>Snow</td>
</tr>
</tbody>
</table>
**SQL Queries - Summary**

**Basic:**

CREATE VIEW $R(B_1, \ldots, B_m)$

SELECT $A_1, \ldots, A_n$

FROM $R_1, \ldots, R_l$

WHERE $C_1 \text{ AND } \ldots \text{ AND } C_o$

**Aggregation:**

SELECT $B_1, \ldots, B_n$, AGGREGATE_OPERATOR($A_i$)

FROM ...

WHERE ...

GROUP BY $B_1, \ldots, B_n$

**Set:**

SQL Query 1

SET_OPERATOR

SQL Query 2

**Nested:**

SELECT ...

FROM ...

WHERE ... AND $A_i$ NEST_OPERATOR (SELECT $A_j$

FROM ...

WHERE ...)