Video Rental DB

<table>
<thead>
<tr>
<th>RID</th>
<th>RDate</th>
<th>CID</th>
<th>CName</th>
<th>MID</th>
<th>MName</th>
<th>Price</th>
<th>Qty</th>
</tr>
</thead>
<tbody>
<tr>
<td>1001</td>
<td>05/15/04</td>
<td>201</td>
<td>John</td>
<td>1</td>
<td>Godfather</td>
<td>$4.50</td>
<td>1</td>
</tr>
<tr>
<td>1001</td>
<td>05/15/04</td>
<td>201</td>
<td>John</td>
<td>2</td>
<td>Goodfellas</td>
<td>$4.50</td>
<td>1</td>
</tr>
<tr>
<td>1001</td>
<td>05/15/04</td>
<td>201</td>
<td>John</td>
<td>3</td>
<td>Casino</td>
<td>$3.50</td>
<td>1</td>
</tr>
<tr>
<td>1002</td>
<td>05/23/04</td>
<td>276</td>
<td>Amy</td>
<td>3</td>
<td>Casino</td>
<td>$3.50</td>
<td>1</td>
</tr>
<tr>
<td>1003</td>
<td>05/23/04</td>
<td>233</td>
<td>Susan</td>
<td>6</td>
<td>You’ve Got Mail</td>
<td>$4.50</td>
<td>1</td>
</tr>
<tr>
<td>1003</td>
<td>05/23/04</td>
<td>233</td>
<td>Susan</td>
<td>7</td>
<td>Kate &amp; Leopold</td>
<td>$4.50</td>
<td>1</td>
</tr>
<tr>
<td>1004</td>
<td>05/24/04</td>
<td>276</td>
<td>Amy</td>
<td>5</td>
<td>Total Recall</td>
<td>$3.50</td>
<td>1</td>
</tr>
</tbody>
</table>

Signs of Bad Design

- Redundancy
- Anomalies
  - Insert
  - Delete
  - Update

Functional Dependency (FD)

- **(A1, A2, ..., An) → (B1, B2, ..., Bm)**
  - If two tuples have the same values of attributes (A1, A2, ..., An), they must have the same values of attributes (B1, B2, ..., Bm)
- **Trivial FD**
  - (B1, B2, ..., Bm) ⊆ (A1, A2, ..., An)
  - (RID) → (RDate, CID, CName)
  - (CID) → (CName)
  - (MID) → (MName, Price)

Keys of a Relation

- **Key**
  - (A1, A2, ..., An) functionally determines all other attributes in the relation
  - **Minimal**
- **Primary Key**
  - Primary key for the Video Rental example??
- **Super Key**

Partial and Transitive Dependency

- **Partial dependency** – a non-key column is dependent on part of the primary key
- **Transitive dependency** – a non-key column is dependent on another non-key column
Dependency Diagram

First Normal Form (1NF)
- Primary key defined
- No duplicate columns or multi-valued columns

```
<table>
<thead>
<tr>
<th>RID</th>
<th>MID1</th>
<th>MID2</th>
<th>MID3</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>2</td>
<td>3</td>
</tr>
</tbody>
</table>
```

Second Normal Form (2NF)
- 1NF
- No partial dependencies

1NF to 2NF
- Separate each partial dependency into a new table
  - Primary key
  - Column

Third Normal Form (3NF)
- 2NF
- No transitive dependencies

2NF to 3NF
- Separate each transitive dependency into a new table
- And ??
Boyce-Codd Normal Form (BCNF)

- Whenever there's a non-trivial FD $(A_1, A_2, ..., A_n) \rightarrow (B_1, B_2, ..., B_m)$ in $R$, $(A_1, A_2, ..., A_n)$ is a super key of $R$.

<table>
<thead>
<tr>
<th>Player</th>
<th>Instrument</th>
<th>Teacher</th>
</tr>
</thead>
<tbody>
<tr>
<td>Chloe</td>
<td>Clarinet</td>
<td>Fred</td>
</tr>
<tr>
<td>Beryl</td>
<td>flute</td>
<td>David</td>
</tr>
<tr>
<td>Kaitlyn</td>
<td>Drums</td>
<td>Christina</td>
</tr>
<tr>
<td>Chloe</td>
<td>flute</td>
<td>David</td>
</tr>
<tr>
<td>Lindsay</td>
<td>flute</td>
<td>Michelle</td>
</tr>
</tbody>
</table>

3NF but not BCNF

Indexes and Constraints

- Indexes – speed up queries
  - B-tree, R-tree, Quad-tree, Hash, Bitmap ...
- Constraints – avoid human errors and speed up queries
  - NOT NULL
  - PRIMARY KEY and UNIQUE
  - FOREIGN KEY and referential integrity

Primary Key And Unique

- **PRIMARY KEY**
  - One or more fields
  - Implies NOT NULL
- **UNIQUE**
  - One or more fields
  - Does not imply NOT NULL
  - Can have more than one UNIQUE constraint

Foreign Key and Referential Integrity

<table>
<thead>
<tr>
<th>Primary Key</th>
<th>Foreign Key</th>
</tr>
</thead>
<tbody>
<tr>
<td>RID</td>
<td>RDate</td>
</tr>
<tr>
<td>Primary Key</td>
<td>CID</td>
</tr>
</tbody>
</table>

- Referenced attribute(s) must be primary key or unique
- Referential integrity
  - A foreign key value is either NULL, or it must exist in the referenced table as a primary or unique key value

Create and Delete Indexes

CREATE [UNIQUE] INDEX index ON table (field [, ...]);

DROP INDEX index ON table;

- **UNIQUE**
- **DISALLOW NULL**
- **IGNORE NULL**
- **PRIMARY**

Not NULL

CREATE TABLE table (
  field type,
  ...
  field type NOT NULL,
  ...
  field type
);
**Primary Key and Unique**

```
CREATE TABLE table (
    field type CONSTRAINT c ( PRIMARY KEY | UNIQUE ),
    ...
);

CREATE TABLE table (  
    field type,
    ...
    CONSTRAINT c ( PRIMARY KEY | UNIQUE ) (field [...])
);
```

**Foreign Key**

```
CREATE TABLE table1 (  
    field type CONSTRAINT c REFERENCES table2 (field),
    ...
);

CREATE TABLE table1 (  
    ...
    CONSTRAINT c FOREIGN KEY (field [...]) REFERENCES table2 (field [...])
);
```