CIS 451/551

Introduction
• relational model
• old alternatives: network model, hierarchical model
• new alternatives: document database, graph database, key-value, column store
• SQL: structured query language
  • from SEQUEL (structured English query language)
• entity-relationship modeling
  • determine a data model
relational model

- database is a collection of *tables* (= *relations*)
- table has *columns* (= *attributes*), each with a data type
- table has *rows* (= *tuples* = *entries*)

- table/columns correspond to the *schema*
- table entries are the data

- database *instance* is an instantiation of a schema
work flow for a database project

1. build ER model
2. convert to relational model
3. enter data
4. build interface (web?) with embedded SQL queries
sample table

<table>
<thead>
<tr>
<th></th>
<th>fname</th>
<th>minit</th>
<th>lname</th>
<th>ssn</th>
<th>bdate</th>
<th>address</th>
<th>sex</th>
<th>salary</th>
<th>superssn</th>
<th>dno</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>John</td>
<td>B</td>
<td>Smith</td>
<td>123456789</td>
<td>1955-01-09</td>
<td>731 Fondren, Houston, TX</td>
<td>M</td>
<td>30000.00</td>
<td>333445555</td>
<td>5</td>
</tr>
<tr>
<td>2</td>
<td>Franklin</td>
<td>T</td>
<td>Wong</td>
<td>333445555</td>
<td>1945-12-08</td>
<td>638 Voss, Houston, TX</td>
<td>M</td>
<td>40000.00</td>
<td>888665555</td>
<td>5</td>
</tr>
<tr>
<td>3</td>
<td>Joyce</td>
<td>A</td>
<td>English</td>
<td>453453453</td>
<td>1962-07-31</td>
<td>5631 Rice, Houston, TX</td>
<td>F</td>
<td>25000.00</td>
<td>333445555</td>
<td>5</td>
</tr>
<tr>
<td>4</td>
<td>Ramesh</td>
<td>K</td>
<td>Narayan</td>
<td>666884444</td>
<td>1952-09-15</td>
<td>975 Fire Oak, Humble, TX</td>
<td>M</td>
<td>38000.00</td>
<td>333445555</td>
<td>5</td>
</tr>
<tr>
<td>5</td>
<td>James</td>
<td>E</td>
<td>Borg</td>
<td>888665555</td>
<td>1927-11-10</td>
<td>450 Stone, Houston, TX</td>
<td>M</td>
<td>55000.00</td>
<td>(null)</td>
<td>1</td>
</tr>
<tr>
<td>6</td>
<td>Jennifer</td>
<td>S</td>
<td>Wallace</td>
<td>987654321</td>
<td>1931-06-20</td>
<td>291 Berry, Bellaire, TX</td>
<td>F</td>
<td>43000.00</td>
<td>888665555</td>
<td>4</td>
</tr>
<tr>
<td>7</td>
<td>Ahmad</td>
<td>V</td>
<td>Jabbar</td>
<td>987987987</td>
<td>1959-03-29</td>
<td>980 Dallas, Houston, TX</td>
<td>M</td>
<td>25000.00</td>
<td>987654321</td>
<td>4</td>
</tr>
<tr>
<td>8</td>
<td>Alicia</td>
<td>J</td>
<td>Zelaya</td>
<td>999887777</td>
<td>1958-07-19</td>
<td>&quot;3321 Castle, Spring, TX&quot;</td>
<td>F</td>
<td>25000.00</td>
<td>987654321</td>
<td>4</td>
</tr>
</tbody>
</table>

*employee table from company database*
terminology

• database
  • collection of data
  • specified format

• database management software (DBMS)
  • software that manages a database or collection of databases
  • MySQL, postgreSQL, Access, Oracle, MariaDB, etc etc
  • adhere to ACID properties

• misuse of terminology
  • common to refer to a DBMS such as MySQL as a “database”
ACID Properties

- Atomicity
- Consistency
- Isolation
- Durability
Maintaining ACID

• Atomicity
  • roll back (related to D, UNDO)

• Consistency
  • up to application programmer, as assumption

• Isolation
  • lock manager

• Durability
  • logs (related to A, REDO)
Table Design

• each table has a PRIMARY KEY
• dbms ensures that the key acts as unique identifier
• only one such entry can exist with that key

• this is *entity integrity*

• important to choose good primary key
• sometimes several choices
CAR and EMPLOYEE example

CAR: license, make, model, year, vin

EMPLOYEE: emp_id, ssn, fname, lname, address

• what are good keys?
• how do we assign cars to employees
• say we use license and emp_id as key
assign CAR to EMPLOYEE options

**option 1**

CAR: license, make, model, year, vin, emp_id

EMPLOYEE: emp_id, ssn, fname, lname, address

Can a car be assigned to many employees?
Can an employee have many cars?
How about here?

option 2

CAR: license, make, model, year, vin

EMPLOYEE: emp_id, ssn, fname, lname, address, license

NOTE:
• when a key to another table is an attribute, it is a FOREIGN KEY
• it is up to a (good) dbms to make sure that the foreign key item exists in the other table
• this is called referential integrity
• this is why we ask you to use the InnoDB storage engine in MySQL, not the MyISAM one
### Bridge table

**option 3**

<table>
<thead>
<tr>
<th>CAR: license, make, model, year, vin</th>
</tr>
</thead>
<tbody>
<tr>
<td>EMPLOYEE: emp_id, ssn, fname, lname, address</td>
</tr>
<tr>
<td>ASSIGN: licence, emp_id, assign_date</td>
</tr>
</tbody>
</table>

There are three choices, depending on the *cardinality* of the CAR::EMPLOYEE relationship.

Another decision: can some of the foreign keys be NULL?
Beginning SQL

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

• order of “execution”
  FROM → WHERE → SELECT