What are Relational Databases?

- Collections of records organized into *tables*

- Tables can be *joined* or *queried* to form new tables

- *Tables* modified dynamically.

Why Relational Databases

- Abstraction between data representation and program

- Existing tools for optimization

- Extensible structure for dealing with data

- Pre-existing administration, backup and recovery

Structured Query Language

- Language for expressing queries and database setup

- Supported by almost all relational database software

- Standardized by ISO and ANSI

What is SQL Like?

- Very verbose

```
group by situation having  count(samples) > 5
    order by sum(samples) desc
```

- Case insensitive

- Comments start with double hyphen

```
-- This is a comment.
```

Some Examples are Oracle-Specific

- SQL is underspecified; things you need to do cannot be stated in the standard language.

- Oracle provides solutions to these problems.

- Oracle can scale to database tasks of almost any size.

- Other databases provide similar solutions.

A Few Definitions

row A single record of related *columns* or *fields*

table A set of *rows*

schema A group of related tables; for example, the library example in this class is a single schema

First, Consider Problem Domain

- Decide on a general area.

- Consider questions to answer/operations to implement.

- What tables/objects are needed?

The Library Example

- We consider the database required to maintain a library.

- Data based loosely on the needs of MITSFS—the MIT Science Fiction Society

- Keep track of books, members, fines, etc

What Might We Be Asked?

- What books do you have by Piers Anthony?

- Who has the copy of *Ender's Game* checked out and when should it be back?

- What's the third book in the *Foundation* series?

- Please check in this book.

Books

- Unique IDs are nice.

- Authors is complex because of multiple authors per book.

- We pretend title is simple.

- We need to store series (single) and position in series.

- Consider useful statistics.

Authors as a Detail Table

- One row in books corresponds to one or more rows in authors.

- A row in authors belongs to exactly one book.

- This relationship is called a master-detail relationship.

- Linked based on unique ID

Members and the Name Problem

- Holds information about members who can potentially borrow books from library.

- Consider producing form letters.

- Other biographic information could also be kept.

- Some members are also librarians.

Copies Table

- Choice for unique ID

- Links to books

- Stores information on who checked out/when due

- Stores circulating status, date obtained, etc

Fines

- Stores who is fined and who is doing the fining

- Stores date issued/paid

- Stores reason and amount

Tables and Datatypes

- A table is defined by its *columns* (fields)

- Each field is usually one of numeric(precision, scale), integer, varchar(length), char(length), or date

- Table creation syntax is finicky

Create Table

```
create table books (
  bookid        integer,
  title         varchar(255),
    -- should really be table like authors
  series        varchar(255),
    -- should really be table like authors
  position      integer,  -- where in series
  checkouts     integer
);

create table authors (
  bookid        integer,
  author        varchar(255)
);
```

Constraints Enforce Data Consistency

- Constraints require that data rows or columns meet certain conditions

- Prevent some forms of inconsistency

- Can be unnamed

```
unique(social_security_num),
phone_number varchar(30) unique
```

- Or named

```
constraint one_ssn_per_cust unique(social_security_num)
```

- Named constraints produce much better errors with Oracle and other databases.

Constraints Not Always the Answer

- Constraints may violate abstraction.

- Constraints may prevent rows from being deleted.

- Not all business rules can be expressed as constraints.

The Horror of Null

- All SQL datatypes can be *null*.

- Null values make most expressions in which they are used null; for example if `a` is null, `a + b` is null.

- The expression `a is null` is true exactly when `a` is null. There is also `a is not null`.

- Nulls represent unspecified or not-available data. For example, the member who is checking out a book copy is null if no one is checking out the copy.

Not-Null Constraints

- If a column can be null, you must check whether the column is null when using it.

- Some data must be specified for software to work correctly. For example, books without titles are useless to a library.

- In these cases, a column can be labeled *not null*.

```
title varchar(255) not null,
```

- You can have named not-null constraints, but unlike other constraints, it doesn't really help.

Not Null Example

```
create table books (
  bookid     integer,
  title     varchar(255) not null,
    --should really be table like authors
  series     varchar(255),
    -- should really be table like authors
  position   integer,  -- where in series
  checkouts  integer not null
);


create table authors (
  bookid    integer not null,
  author    varchar(255) not null
);
```

Unique Constraints

- Having two members with the same ID would be bad; other similar situations arrise.

- A unique constraint requires that a column or set of columns be unique within a table. If you have multiple columns in a unique constraint, the composite of all values must be unique.

- Unique columns can be null.

- Example:

```
constraint  one_social_security_num unique(social_security_num)
```

Unique Example

```
create table books (
  bookid     integer
   constraint books_primary unique(bookid),
  title     varchar(255) not null,
    --should really be table like authors
  series     varchar(255),
    -- should really be table like authors
  position   integer,  -- where in series
  checkouts  integer not null
);

create table authors (
  bookid    integer,
  author    varchar(255),
   constraint authors_primary unique(bookid, author)
);
```

Primary Key

- Previous example was lacking, because columns could have been null.

- Column values must be unique and non-null

- Example:

```
constraint  books_primary primary key(bookid)
```

- Alternatively:

```
bookid integer
    constraint books_primary primary key,
```

Primary Key Example

```
create table books (
  bookid     integer,
  constraint bookid_primary primary key(bookid),
  title     varchar(255) not null,
    --should really be table like authors
  series     varchar(255),
    -- should really be table like authors
  position   integer,  -- where in series
  checkouts  integer not null
);

create table authors (
  bookid    integer,
  author    varchar(255),
  constraint authors_primary primary key (bookid, author)
);
```

Foreign Keys

- In a master-detail relationship, it is important to make sure that the detail record refers to a real master record. A foreign key constraint enforces this by requiring that a specific column or column group appears as in another table.

- Foreign keys must map to unique or primary keys in the referencing table.

- Example:

```
constraint fines_librarian_exists foreign key(librarian)
    references members(memberid)
```

- The foreign key can be null unless otherwise constrained.

Foreign Key Examples

```
create table books (
  bookid      integer
    constraint bookid_primary primary key,
  title       varchar(255) not null,
    --should really be table like authors
  series      varchar(255),
    -- should really be table like authors
  position    integer  -- where in series
);

create table authors (
  bookid      integer
  constraint authors_bookid_exists
    foreign key(bookid) references books(bookid),
  author      varchar(255),
  constraint authors_primary primary key (bookid, author)
);
```

Check Constraints

- Sometimes, constraints must be specified using general SQL expressions.

- Check constraints allow this.

- Most databases require the expression reference only the current row and not reference run-time state like the system date.

- Example:

```
constraint circ_valid check(circ in (0, 1))
```

Check Example

```
create table books (
  bookid      integer
    constraint bookid_primary primary key,
  title       varchar(255) not null,
    --should really be table like authors
  series      varchar(255),
  position    integer,  -- where in series
  checkouts   integer not null   ,
  constraint position_meaningful
    check(position is null or position >= 0)
);
create table authors (
  bookid      integer
  constraint authors_bookid_exists
    foreign key(bookid) references books(bookid),
  author      varchar(255),
  constraint authors_primary primary key (bookid, author)
);
```

Default Values

- A default clause can be added to columns to specify a default value.

- Because of an Oracle bug, do not include comments after default clauses; doing so can corrupt database backups. They are OK after the comma that closes the field or the paren that closes the create table, but not before.

- Example:

```
checkouts integer not null default 0
```

Default Example

```
create table books (
  bookid      integer
    constraint bookid_primary primary key,
  title       varchar(255) not null,
  series      varchar(255),
  position    integer,  -- where in series
  checkouts   integer not null  default 0,
  constraint position_meaningful
    check(position is null or position >= 0)
);
create table authors (
  bookid      integer
  constraint authors_bookid_exists
    foreign key(bookid) references books(bookid),
  author      varchar(255),
  constraint authors_primary primary key (bookid, author)
);
```

Alter Table

- You can add a column after a table is created:

  ```
  alter table copies add isbn integer;
  ```

- Table constraints can also be added.

- Implementations including Oracle allow column data types and null constraints to be modified. There is not a standard SQL manner for doing this.

Drop Table

- The **`drop table`** command can be used to delete a table and all contained data.

- Tables may not be dropped if this would violate a foreign-key constraint.

- Oracle provides the **`cascade constraints`** operation to delete all constraints needed to allow the drop to happen.

- Example:

```
drop table books cascade constraints;
```