

Databases

Why?

- Abstraction of logical from physical structure
- Allows separation of a program's "business logic" from concerns about traversal of the data

Types of databases

- Object
 - direct representation of programming language objects
- Relational (<== dominant)
 - Tables
 - Operations
 - Select, project, union
 - Join (natural, inner, outer, left, right, ...)
 - Indexes
- Hierarchical (e.g., XML)
 - Parent-child
- Network
- Flat files (e.g., spreadsheet, text file)

Hierarchical Model

Figure from Association for Computing Machinery removed due to copyright restrictions.
See Levin, Michael. "An introduction to DIAM: levels of abstraction in accessing information." *Association for Computing Machinery*, 1978.

Network Model

Figure from Association for Computing Machinery removed due to copyright restrictions.
See Levin, Michael. "An introduction to DIAM: levels of abstraction in accessing information." *Association for Computing Machinery*, 1978.

Relational Database Underlying Concepts

- Individual entities
- Their properties
- Relations among them
 - 1-1
 - 1-n (or n-1)
 - n-n
- Integrity
- Transactions

Relational Model

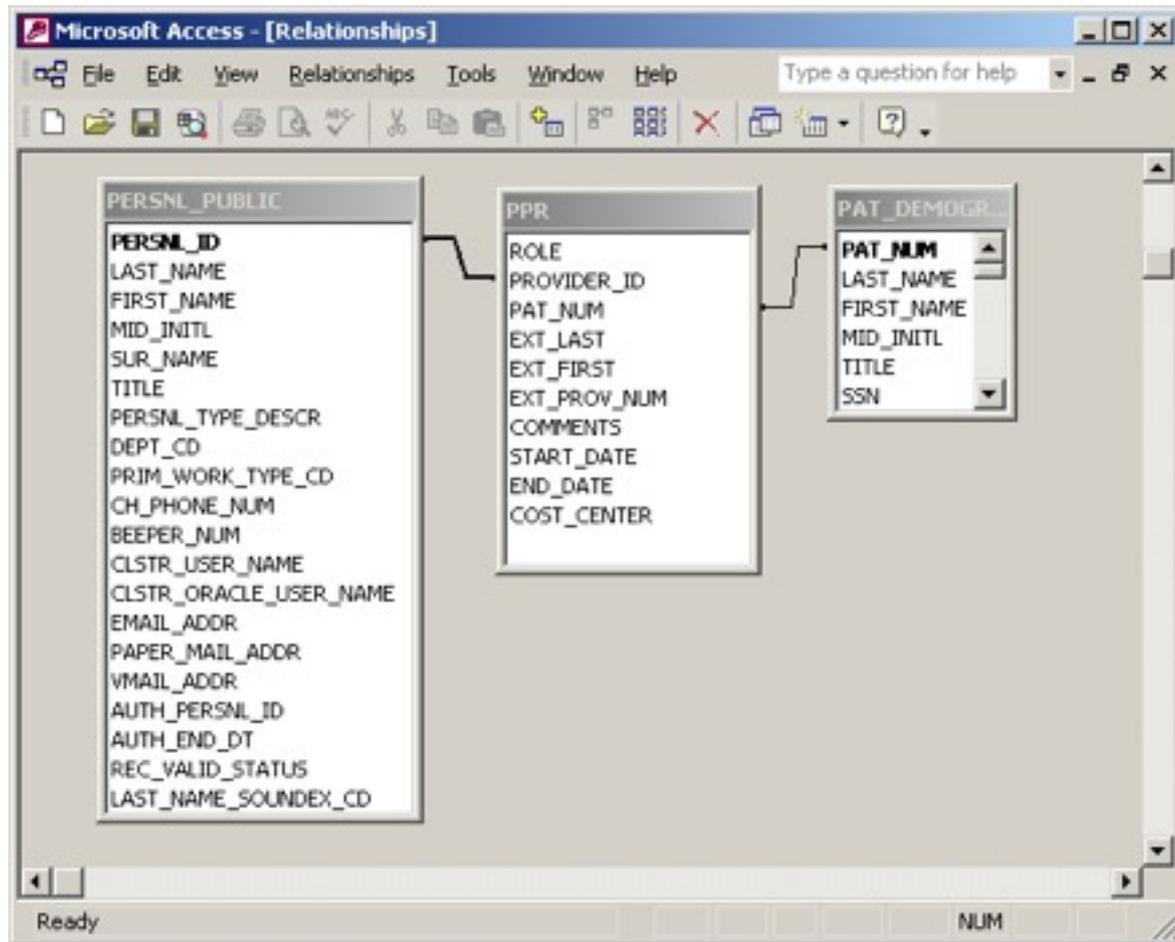
Figure from Association for Computing Machinery removed due to copyright restrictions.
See Levin, Michael. "An introduction to DIAM: levels of abstraction in accessing information." *Association for Computing Machinery*, 1978.

Relational Algebra Operations

- Select—subset of rows with conditions
- Project—subset of columns
- Join A and B
 - Outer: cross product of all rows in A and B, result includes all columns of each
 - Natural: select rows of cross-product in which matching columns have same values
 - Join on specific column relations ($=$, $>$, $<$, ...)
- Grouping operations (partition by criteria)
- Summarization (count, max, min, average)

MySQL SELECT syntax

```
SELECT [ALL | DISTINCT | DISTINCTROW ]
       [HIGH_PRIORITY] [STRAIGHT_JOIN] [SQL_SMALL_RESULT]
       [SQL_BIG_RESULT] [SQL_BUFFER_RESULT] [SQL_CACHE | SQL_NO_CACHE]
       [SQL_CALC_FOUND_ROWS]
       select_expr, ...
       [FROM table_references
        [WHERE where_condition]
        [GROUP BY {col_name | expr | position}
         [ASC | DESC], ... [WITH ROLLUP]]
        [HAVING where_condition]
        [ORDER BY {col_name | expr | position}
         [ASC | DESC], ...]
        [LIMIT {[offset,] row_count | row_count OFFSET offset}]
        [PROCEDURE procedure_name(argument_list)]
        [INTO OUTFILE 'file_name' export_options
         | INTO DUMPFILE 'file_name'
         | INTO @var_name [, @var_name]]
        [FOR UPDATE | LOCK IN SHARE MODE]]
```



MySQL SELECT examples

```
select * from persnl_public where last_name='Bird';
```

```
select pat_num from persnl_public, ppr
  where persnl_public.persnl_id=ppr.provider_id
  and persnl_public.last_name='Bird';
```

```
select d.last_name,d.first_name
  from persnl_public as p, ppr, pat_demograph as d
  where p.persnl_id=ppr.provider_id
  and ppr.pat_num=d.pat_num
  and p.last_name='Bird';
```

```
select p.last_name,p.first_name,count(*) as c
  from persnl_public as p join ppr
    on p.persnl_id=ppr.provider_id
  group by p.persnl_id
  having c>1
  order by c desc;
```

MIT OpenCourseWare
<http://ocw.mit.edu>

HST.950J / 6.872 Biomedical Computing
Fall 2010

For information about citing these materials or our Terms of Use, visit: <http://ocw.mit.edu/terms>.