Views

- A virtual table consists of the results of a query.
- Example: create a view
  
  `members_salespeople_view`
  
  `(member_name, salesperson_name)`

About Views

- The data in a view is dynamically computed.
  - Changes to base tables are automatically reflected in the view.
- A view can be used as a table in SQL queries.
- Views cannot be updated (except in some very rare cases).

Why Views

- Present the data in a different way.
- Simplify SQL queries.
- Security reasons.
  - E.g. expose only part of the data to certain type of users.

Indexes

- Make query execution more efficient.

Query Example

```
select salary from employees where name = 'Sally';
```

<table>
<thead>
<tr>
<th>employees</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>name</td>
<td>salary</td>
</tr>
<tr>
<td>Joe</td>
<td>2000</td>
</tr>
<tr>
<td>Bob</td>
<td>5000</td>
</tr>
<tr>
<td>Lisa</td>
<td>4000</td>
</tr>
<tr>
<td>Amy</td>
<td>4500</td>
</tr>
<tr>
<td>John</td>
<td>4500</td>
</tr>
<tr>
<td>Sally</td>
<td>5000</td>
</tr>
<tr>
<td>Val</td>
<td>3000</td>
</tr>
<tr>
<td>Meg</td>
<td>6000</td>
</tr>
</tbody>
</table>
Search with an Index

Create Index

```
create index index_name on table_name (col_name, ...);
```

Example: create an index on the name column of the employees table

```
create index emp_name_idx on employees (name);
```

The Need for Transaction

Example: transfer $100 from account A to account B

<table>
<thead>
<tr>
<th>accounts</th>
<th>balance</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>134.60</td>
</tr>
<tr>
<td>B</td>
<td>122.21</td>
</tr>
<tr>
<td>C</td>
<td>3300.00</td>
</tr>
<tr>
<td>D</td>
<td>256.79</td>
</tr>
</tbody>
</table>

SQL Statements Involved in A Transfer

```
-- Check whether account A has enough money
select balance from accounts where account = 'A';

-- Take $100 from account A
update account set balance = balance - 100
where account = 'A';

-- Add $100 to account B
update account set balance = balance + 100
where account = 'B';
```

Things Could Go Wrong

```
-- Check whether account A has enough money
select balance from accounts where account = 'A';

-- Take $100 from account A
update account set balance = balance - 100
where account = 'A';
```

System Crash!

Transaction

A group of statements that are treated as a whole, i.e. either all operations in the group are performed or none of them are — the Atomicity property.
Transaction Syntax in MySQL

begin; -- start of a transaction
select balance from accounts where account = 'A';
update account set balance = balance - 100
where account = 'A';
update account set balance = balance + 100
where account = 'B';
commit; -- end of a transaction
(or rollback;)

ACID Properties of Database Transactions

- Atomic
- Consistent
- Isolated
- Durable