DB modifications, plus transactions

Modifying data with SQL

To maintain the courses data, you'd need to:

- add new courses
- delete old instructors ☹
- increase instructors' salaries ☺
- etc.

Inserting rows into a table

We have already seen simple inserts, like this:

```
create table department
(
  dept_name varchar(20) primary key,
  building varchar(15) not null,
  budget integer not null
);

insert into department values("Biology", "Watson", 90000);
insert into department values("Comp. Sci.", "Taylor", 100000);
insert into department values("Elec. Engr.", "Taylor", 85000);
```

Using attribute names when inserting

```
create table department
(
  dept_name varchar(20) primary key,
  building varchar(15) not null,
  budget integer not null
);
```

basic version

```
insert into department
        values("Elec. Engr.", "Taylor", 85000);
```

use attribute names

```
insert into department(dept_name, building, budget)
        values("Biology", "Watson", 90000);
```

reorder attribute names

```
insert into department(building, dept_name, budget)
        values(“Taylor”, "Comp. Sci.”, 100000);
```

Using queries when inserting

student

ID	name	dept_name	tot_cred
128	Zhang	Comp. Sci.	102
12345	Shankar	Comp. Sci.	32
19991	Brandt	History	80
23121	Chavez	Finance	110
15151	Spears	Music	150
45678	Levy	Physics	46

instructor

ID	name	dept_name	salary
10101	Srinivasan	Comp. Sci.	65000
12121	Wu	Finance	90000
15151	Spears	Music	18000
22222	Einstein	Physics	95000
32343	El Said	History	60000
33456	Gold	Physics	87000
45565	Katz	Comp. Sci.	75000
58583	Califieri	History	62000
76543	Singh	Finance	80000
76766	Crick	Biology	72000
83821	Brandt	Comp. Sci.	92000
98345	Kim	Elec. Eng.	80000

```
insert into instructor
        select ID, name, dept_name, 18000
        from student
        where dept_name = “Music” and tot_cred > 144;
```

Deleting rows from a table

instructor

ID	name	dept_name	salary
10101	Srinivasan	Comp. Sci.	65000
12121	Wu	Finance	90000
15151	Spears	Music	18000
22222	Einstein	Physics	95000
32343	El Said	History	60000
33456	Gold	Physics	87000
45565	Katz	Comp. Sci.	75000
58583	Califieri	History	62000
76543	Singh	Finance	80000
76766	Crick	Biology	72000
83821	Brandt	Comp. Sci.	92000
98345	Kim	Elec. Eng.	80000

```
delete from instructor
where name = “Spears”;
```

```
delete from instructor
where dept_name = Physics
    and salary > 80000;
```

```
delete from instructor
where dept_name in
   (select dept_name
       from department
       where building = “Watson”);
```

no ‘select’ part is allowed – why?

Updating tuples

Question:

If there were no SQL support for updating a table, could you live without it?

Yes – delete the rows you want to update, then insert new ones

Updating rows of a table

instructor

ID	name	dept_name	salary
10101	Srinivasan	Comp. Sci.	65000
12121	Wu	Finance	90000
15151	Spears	Music	18000
22222	Einstein	Physics	95000
32343	El Said	History	60000
33456	Gold	Physics	87000
45565	Katz	Comp. Sci.	75000
58583	Califieri	History	62000
76543	Singh	Finance	80000
76766	Crick	Biology	72000
83821	Brandt	Comp. Sci.	92000
98345	Kim	Elec. Eng.	80000

```
update instructor
set salary = 70000
where name = “Califieri”;
```

```
update instructor
set salary = salary * 1.05
where salary < 70000;
```

```
update instructor
set salary = salary * 2.0,
    name=“Platinum”
where name=“Gold”;
```

```
update instructor
set salary = case
   when salary < 60000 then
        salary * 1.1
   else salary * 1.05
;
```

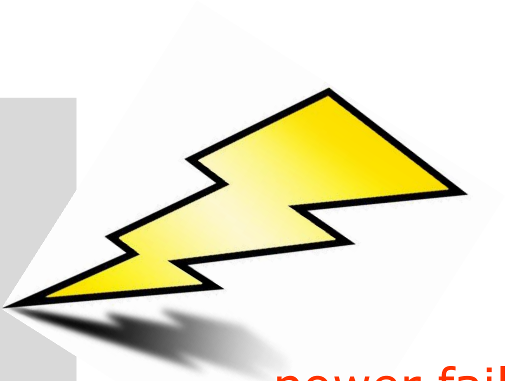
Transactions: why?

Scenario 1: bank funds transfer

account_id	name	balance
101	Rene Wells	4250.55
102	Janie Abbott	34.88
103	Andrea Owen	955.20
104	Javier Benson	772.59

To transfer $60.00 from Rene to Andrea:

1. Decrease balance of Rene by $60.00
2. Increase balance of Andrea by $60.00



power failure after step 1, before step 2!

Transactions in SQL

account_id	name	balance
101	Rene Wells	4250.55
102	Janie Abbott	34.88
103	Andrea Owen	955.20
104	Javier Benson	772.59

```
begin;
  update account
     set balance = balance - 60
   where account_id = 101;
  update account
     set balance = balance + 60
   where account_id = 103;
commit;
```

If an error occurs during the transaction, SQLite will attempt to:

- redo current update
- finish the transaction