Triggers

What is a trigger?

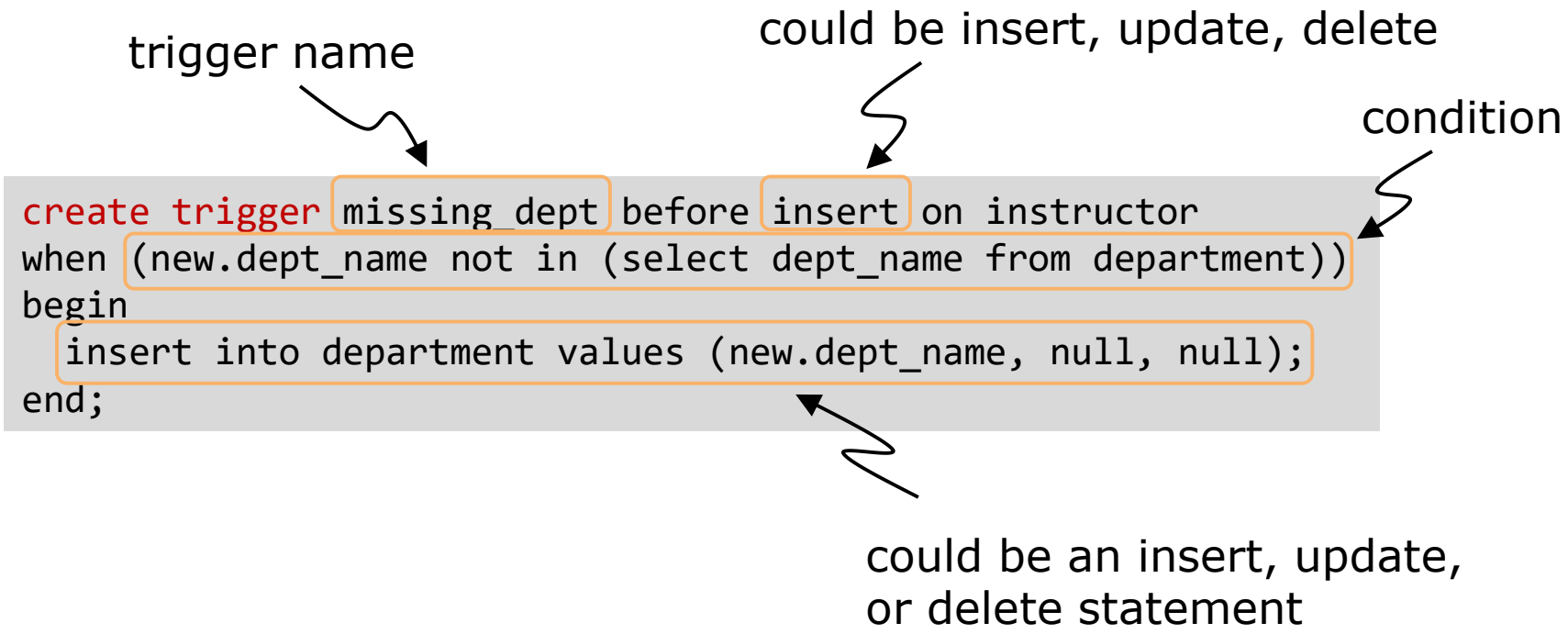
A trigger is something that executes when a DB operation is performed

```
create trigger missing_dept before insert on instructor
when (new.dept_name not in (select dept_name from department))
begin
  insert into department values (new.dept_name, null, null);
end;
```

This says:

- when an instructor is inserted
- if the dept_name is not in the dept_name table
- then add a new dept in the dept_name table

Parts of a trigger



Different database systems have different forms of trigger statements. SQLite trigger statements differ from what's in the text.

Trigger execution

```
sqlite> create trigger missing_dept before insert on instructor
   ...> when (new.dept_name not in (select dept_name from department))
   ...> begin
   ...>   insert into department values (new.dept_name, null, null);
   ...> end;
sqlite> insert into instructor values(74839, "Ranklin", "Phys. Ed.", 58000);
sqlite> select * from department;
Biology|Watson|90000
Comp. Sci.|Taylor|100000
Elec. Eng.|Taylor|85000
Finance|Painter|120000
History|Painter|50000
Music|Packard|80000
Physics|Watson|70000
Phys. Ed.||
```

Trigger example: enforce a constraint

```
sqlite> create trigger timeslot_check before insert on section
   ...> when (new.time_slot_id not in (select time_slot_id from time_slot))
   ...> begin
   ...>   select raise(abort, 'no such time slot in time_slot table');
   ...> end;
sqlite>
sqlite> insert into section values("CS-101","2","Fall",2010,"Packard","201","J");
Error: no such time slot in time_slot table
```

This approach allows custom error handling of constraint violations

Trigger example: repair a bad insert

```
sqlite> create trigger set_null after insert on takes
   ...> when (new.grade="")
   ...> begin
   ...>   update takes set grade = null
   ...>   where id=new.ID and course_id=new.course_id and sec_id=new.sec_id and
   ...>         semester=new.semester and year=new.year;
   ...> end;
sqlite>
sqlite> insert into takes values(55739, "CS-101", "1", "Spring","2010","");
sqlite> select * from takes where course_id = "CS-101";
00128|CS-101|1|Fall|2009|A
12345|CS-101|1|Fall|2009|C
45678|CS-101|1|Fall|2009|F
45678|CS-101|1|Spring|2010|B+
54321|CS-101|1|Fall|2009|A-
76543|CS-101|1|Fall|2009|A
98765|CS-101|1|Fall|2009|C-
55739|CS-101|1|Spring|2010|
```

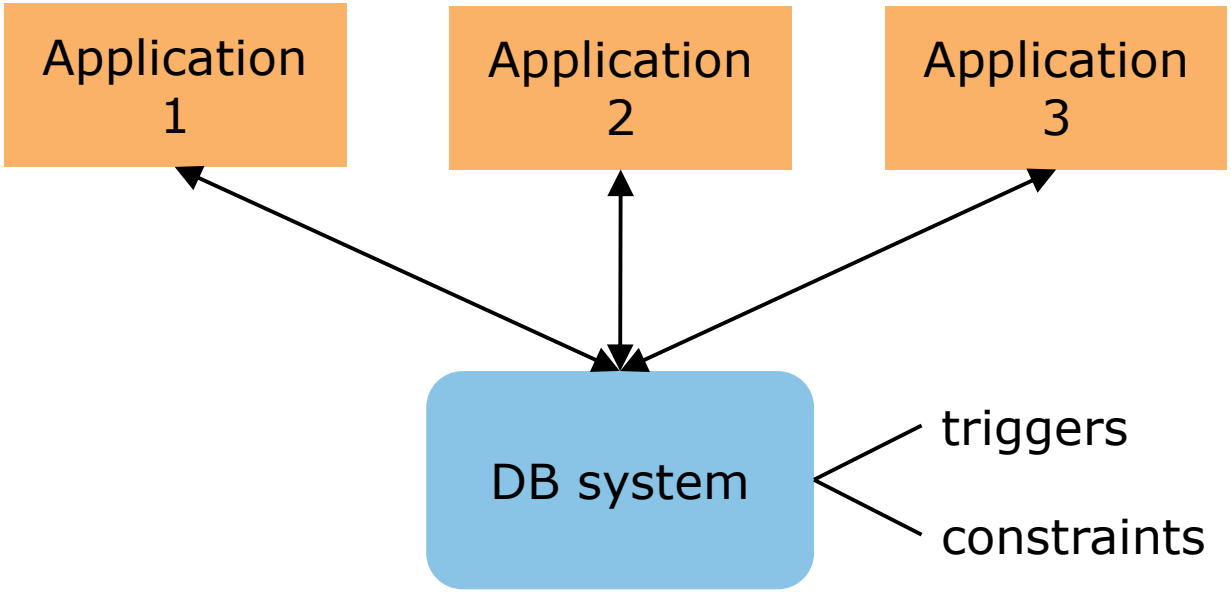
If someone entered “” for grade, they probably meant ‘null’

Trigger example: write to audit table

```
sqlite> create table salary_audits
   ...> (
   ...>         ID   varchar(5),
   ...>         name varchar(20),
   ...>         dept_name varchar(20),
   ...>         old_salary numeric(8,2),
   ...>         new_salary numeric(8,2),
   ...>         time_of varchar(25)
   ...> );
sqlite> create trigger salary_update after update on instructor
   ...> when (new.salary > old.salary * 1.1)
   ...> begin
   ...>         insert into salary_audits
   ...>         values(new.ID, new.name, new.dept_name, old.salary,
   ...>                new.salary, datetime('now'));
   ...> end;
sqlite> update instructor set salary=120000 where name="Mozart";
sqlite> select * from salary_audits;
15151|Mozart|Music|40000|120000|2015-09-21 18:09:42
```

People should manually review suspicious salary changes

Moving application logic to the DB



What's great about constraints and triggers is that they are defined once, in the DB system.

Not separately in each application.

Problems with triggers

- The can be hard to understand
  - especially when one trigger triggers another trigger
- The DB can’t use them in optimizations
- They are procedural, not declarative

When to use triggers

- basic rule: don't use a trigger if a constraint can be used instead
  - Constraints are in the "declarative" spirit of SQL
- Triggers can be used when a integrity constraint can't be expressed in SQL
- Triggers are often also used with audit tables
  - better than having each application do auditing

Triggers in SQLite

- viewing triggers
  - for one table, shown with schema
  - `select * from sqlite_master where type = 'trigger';`
- deleting a trigger
  - `drop trigger missing_dept;`