## Deletion

- Can only delete a tuple/tuples. Cannot delete values for particular attibutes

  `delete from` $r$
  `where` $P$

  Deletes all tuples $t$ in $r$ for which $P(t)$ is true.

- `delete` operates on **one relation only**.

- `delete from r` deletes all tuples (should get a warning; not in MySQL!!).

- delete employees with `null` salary

  `delete from` *employee*
  `where` *salary* `is null`;

- delete employees from department with dept_id 4: This will not work:

  `delete from` *employee*
  `where` *name* `in` (`select` *name*
  `from` *employee*
  `where` $dept\_id = 4$);

  Why? (Delete and Updates in MySQL do not allow the relation which is to be updated to appear in the `WHERE` clause). Solution: rename the `employee` relation.

  `delete from` *employee*
  `where` *name* `in` (`select` *name*
  `from` (`select name`
  `from` *employee*
  `where` $dept\_id = 4$) `as` $x$);

## Insertion

- *Without specifying attributes.*

  `insert into` *employee*
  values $(`Verne', 2400.00, 4, 1, `Systems\ Programmer')$;

- *With attributes specification.*

  `insert into` *employee* (name, dept_id, title, salary, cat_id)
  values $(`Verne', 4, `Systems\ Programmer', 2400.00, 1)$;

  If an attribute is not specified, `null` value is inserted for it in the inserted tuples.

- Can also use the result of a query as the tuples for insertion.
  - A new relation:

    customer(c_name, dept_id)

    ```
    create table customer (c_name char(20), dept_id int(20));
    ```

  - Assume that all employees not working in department with id 2 are customers of the company.

    ```
    insert into customer
    select name, dept_id
    from employee
    where dept_id <> 2;
    ```

- **Insertion for views.** If a tuple(s) is inserted in a view, they are actually inserted in the source relation and any missing attributes are given `null` values. For e.g.

  ```
  insert into exp_employees values('Neal', 3120.00);
  ```

  will insert the values `('Neal', 3120.00, NULL, NULL, NULL)` in `employee`, and a corresponding tuple `('Neal', 3120.00)` appears in `exp_employees`. Often a tuple inserted in the view may not appear in the view itself if it does not satisfy the conditions.

  ```
  insert into exp_employees values('Case', 3120.00);
  ```

  will not appear in `exp_employees`, but will appear in the `employee` relation.

## Updates

- For updating only some attribute values for a tuple.

  ```
  update employee
  set salary = 2000.00;
  ```

  ```
  update employee
  set salary = salary * 1.05;
  ```

- Conditional updates:

  ```
  update employee
  set salary = salary * 1.05
  where salary ≤ 2000;
  ```

```
update employee
set salary = 4000.00, dept_id = 2, title='CEO'
where name = 'Yanyen';
```

Trying to update a relation with the same relation occuring in the `where` clause; this won't work:

```
update employee
set salary = salary * 1.05
where salary ≤ (select avg(salary)
from employee);
```

this will:

```
update employee
set salary = salary * 1.05
where salary ≤ (select avg(salary)
from (select salary
from employee  as temp);
```

- Following two statements accomplish: "Give a 5% raise to those with salary less than 3000 and a 2% raise to those earning less than 3000."

```
update employee
set salary = salary * 1.05
where salary  < 3000;

update employee
set salary = salary * 1.02
where salary  > 3000;
```

To combine it into one, we can use the **case** statement:

```
update employee
    set salary = case
        when salary  < 3000 then salary ∗ 1.05
        else salary  ∗ 1.02
     end
```

General form of the **case** statement is:

```
case
    when pred_i then result_i
    when pred_j then result_j
    ...
    when pred_n then result_n
 end
```

- **Updating a View.** A view may be updated, which essentially updates the source relation.
  - Suppose we have the view created by:

    ```
    create view exp_employees as
    select name, salary from employee
    where salary > 4000;
    ```

  - Updating the view using the following statement will update the `employee` relation which will in turn update the `exp_employees` view:

    ```
    update exp_employee
    set salary = 5200
    where name = 'Amit'
    ```