# Migrations and Misc Rails Reference Cheat Sheet

by dwapi via cheatography.com/43665/cs/13036/

## ORM (Object Relational Mapping)

### Advantages

- DB independent. moving from PG to Sqlite should be trivial.
- Cleaner code focused on business logic rather than SQL Query
- Relationship/Association Management
- Model Objects can be cached -- No need to repeatedly query DB for same data
- Transaction support
- Built in validation before save

### Disadvantages

- Abstraction = overhead (uses more memory etc...)
- SQL ignorance

## Create Table (with Datatypes)

```
# CREATE
class Create Orders < Active Rec ord ::M igr ati -
on[5.1]
   def change
        cre ate _table :orders do |t|

              t.b elo ngs_to :category
              t.b inary :photo # PG: bytea MySQL:
blob
              t.b oolean :success # PG: boolean
MySQL: tinyint(1)
              t.date :ship_date # PG & MySQL: date
              t.d atetime :order _date # PG:
timestamp MySQL: datetime
             # PG & MySQL: decimal Precise. Use
for currency, geoloc ation
              t.d ecimal :latitude, :decimal,
:precision => 15, :scale => 13
              t.float # PG & MySQL: float Less
precise thank decimal
              t.i nteger :age # PG: integer MySQL:
int(11)
              t.s tring :name # PG & MySQL:
varcha r(255)
              t.text :comments # PG & MySQL: text
              t.time :ship_time # PG & MySQL: time
              t.t ime stamps # PG: timestamp
MySQL: datetime
       end
   end
end
```

## Generators

rails g model comment post:references text:string

rails g migration add_post_to_user post:belongs_to

rails g migration AddUserRefToProducts user:references

rails g migration AddPartNumberToProducts part_number:string

rails generate controller Sessions new

## rails db: command

**Rollback the last migration:**

rails db:rollback

**Revert the last 3 migrations:**

rails db:rollback STEP=3

**Rollback and then migrating back up:**

rails db:migrate:redo STEP=3

**Run specific migration**

rails db:migrate:up VERSION=20080906120000

**Load test fixtures**

rails db:fixtures:load

**Create a db/schema.rb file that can be portably used against any database supported by ActiveRecord**

db:schema:dump

**Load a schema.rb file into the database**

db:schema:load

**Dump database structure to SQL file Drops the database, creates the datab**

db:structure:dump

**Clone your database structure into the test database**

db:test:prepare

**Drops the database, creates the database and then runs migra- tions against the database. Takes a VERSION argument as well as RAILS_ENV**

db:reset

**Create/clear Sessions table**

db:sessions:create (or :clear)

## Postgres Specific Data Types

```
:hstore # Key => Val Hash Store
:json #The json data type stores an exact copy of
the input text
:jsonb # stored in a decomposed binary format
:array
:cidr_ address #IP Address
:ip_ad dress #IP Address
:mac_a ddress #String
```

In other DBs these are all stored as strings

## Change Table

```
class AddPartNumberToProducts <
ActiveRecord::Migration[5.0]
   def change

        add _index :products, :part_ number
        rem ove _index :products, :part_ number
        add _re ference :this_ table, :other -
_table, index: true
        rem ove _re ference :this_ table, :other -
_table, index: true

        bel ong s_to:
        add _column :products, :part_ number,
:string
        add _column :products, :colum n_name,
:string, :limit => 50
        ren ame _column :shoes, :season, :season_id
        t.c hange :colum n_name, :new_c olu mn_type
        cha nge _co lum n_d efault :table _name,
:status, 0
        t.r emove :colum n_name

   end
end
```