* SHOW TABLES;
* CREATE TABLE Persons (PersonID int,LastName varchar(255),FirstName varchar(255),Address varchar(255),City varchar(255));
* CREATE TABLE person(id INT, fname varchar(255), lname varchar(255), PRIMARY KEY (id));
* CREATE TABLE person(id INT PRIMARY KEY, fname varchar(255), lname varchar(255), FOREIGN KEY (id) REFERENCES student(id));
* CREATE TABLE person(id INT, fname varchar(255), lname varchar(255), UNIQUE(id));
* CREATE TABLE person(id INT, fname varchar(255) NOT NULL, lname varchar(255) NOT NULL);
* ALTER TABLE Persons ADD email varchar(255);
* ALTER TABLE Persons DROP COLUMN email;
* ALTER TABLE Persons ADD CONSTRAINT em\_un UNIQUE (PersonID,LastName);
* ALTER TABLE users ADD CONSTRAINT email\_unique UNIQUE (email);
* ALTER TABLE Persons DROP CONSTRAINT em\_un;
* ALTER TABLE Persons RENAME TO Person;
* ALTER TABLE Person RENAME COLUMN LastName TO lname;
* DROP TABLE Persons;
* TRUNCATE TABLE Person;
* INSERT INTO person(id,fname,lname) VALUES(1,'joey','tribbianni');
* INSERT INTO person(id,fname,lname) VALUES(2,'phoebe','buffay'),(3,'ross','geller');
* INSERT INTO person(id,fname,lname) SELECT id,firstname,lastname FROM student;
* UPDATE person SET fname = 'pheebs' WHERE id = 22;
* UPDATE person SET fname = 'pheebs';
* DELETE FROM person;
* DELETE FROM person WHERE id=1;
* CREATE VIEW personview(id,name) AS SELECT id,fname FROM Person;
* CREATE VIEW pview(id,name) AS SELECT id,fname FROM Person WITH CASCADED CHECK OPTION;
* CREATE VIEW pview(id,name) AS SELECT id,fname FROM Person WITH LOCAL CHECK OPTION;
* DROP VIEW pview;
* CREATE INDEX p\_idx ON person(id,fname);
* CREATE UNIQUE INDEX pe\_idx ON person(fname,lname);
* ALTER TABLE person DROP INDEX p\_idx;
* SELECT \* FROM person;
* SELECT id,fname FROM person;
* SELECT fname FROM person ORDER BY id ASC;
* SELECT fname FROM person ORDER BY id DESC;
* SELECT fname AS name FROM person;
* SELECT co.firstname, ci.fname FROM person AS ci JOIN student AS co ON ci.id = co.id;
* SELECT lname FROM person WHERE id > 3;
* SELECT fname FROM person WHERE fname != 'ross' AND fname != 'John';
* SELECT lname FROM person WHERE lname LIKE 'g%' OR lname LIKE '%y';
* SELECT fname,lname FROM person WHERE lname LIKE '\_eller';
* SELECT fname FROM person WHERE id BETWEEN 3 AND 7;
* SELECT fname FROM person WHERE lname IS NOT NULL;
* SELECT fname FROM person WHERE id IN (1, 4, 7, 8);
* SELECT person.fname, student.firstname FROM person INNER JOIN student ON person.id = student.id;
* SELECT person.lname, student.firstname FROM person LEFT JOIN student ON person.id = student.id;
* SELECT person.lname, student.firstname FROM person RIGHT JOIN student ON person.id = student.id;SELECT person.lname, student.firstname FROM person CROSS JOIN student;
* SELECT person.lname, student.firstname FROM person NATURAL JOIN student;
* SELECT COUNT(\*) FROM person;
* SELECT COUNT(id) FROM person;
* db.student.insertOne({"name": "Max"});
* db.student.insertMany([{"name": "Max"}, {"name":"Alex"}]);
* db.student.insertMany([{"name": "Max"}, {"name":"Alex"}], {"ordered":"false"});
* db.student.insertOne({"date": ISODate()});
* db.student.findOne({"name": "Max"});
* db.student.find({"name": "Max"});
* db.student.find({"name": "Max"}).pretty();
* db.student.find({"name": "Max", "age": 32});
* db.student.distinct("name");
* db.student.countDocuments({"age": 32});
* db.student.estimatedDocumentCount();
* db.student.find({"year": {"$gt": 1970}});
* db.student.find({"name":{"$not": {"$eq": "Max"}}});
* db.student.find({"$or": [{"year" : 1951}, {"year" : 1959}]});
* db.student.updateOne({"\_id": "65962cdfc89264c2da745f01"}, {"$set": {"year": 2016, "name": "Maxz"}});