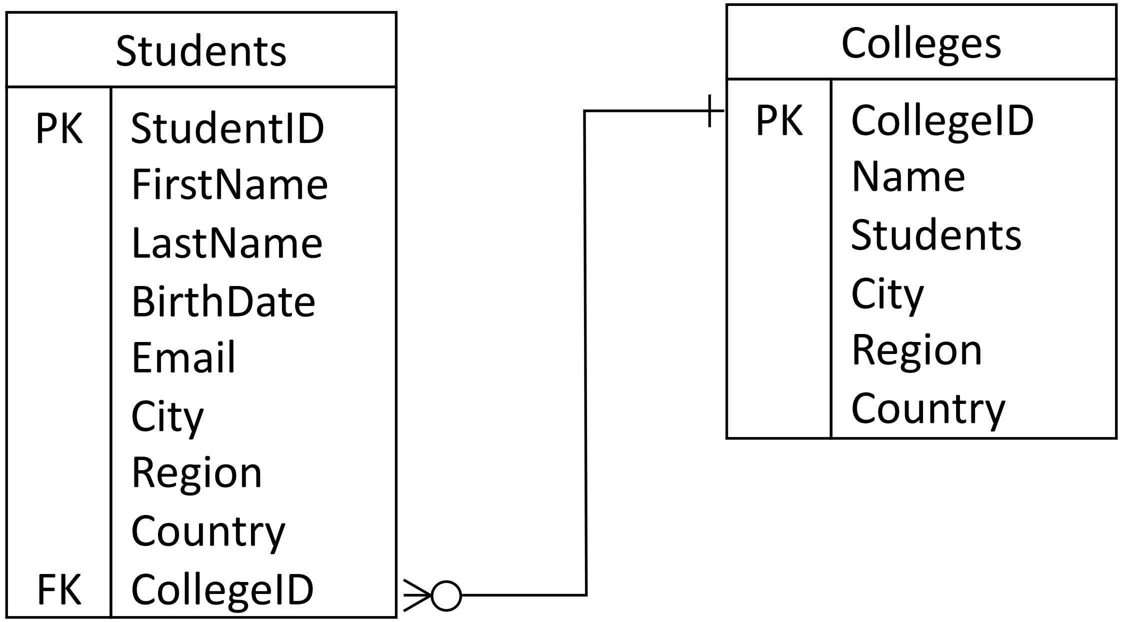
Creating and populating a database

Education Database



Students	
PK	StudentID
	FirstName
	LastName
	BirthDate
	Email
	City
	Region
	Country
FK	CollegeID

Colleges	
PK	CollegeID
	Name
	Students
	City
	Region
	Country

Creating a databases

CREATE DATABASE ***DatabaseName***

Code example:

```
CREATE DATABASE IF NOT EXISTS `education`;
```

Databases on the server

SHOW DATABASES

Removing a database

DROP DATABASE *DatabaseName*

Code example:

```
DROP DATABASE IF EXISTS `education`;
CREATE DATABASE IF NOT EXISTS `education`;
```

Choosing a database

USE ***DatabaseName***

Code example:

```sql
DROP DATABASE IF EXISTS `education`;
CREATE DATABASE IF NOT EXISTS `education`;
USE `education`;

SET NAMES UTF8MB4;
SET character_set_client = UTF8MB4;
```

* In the GUI, you can also double click on the database name

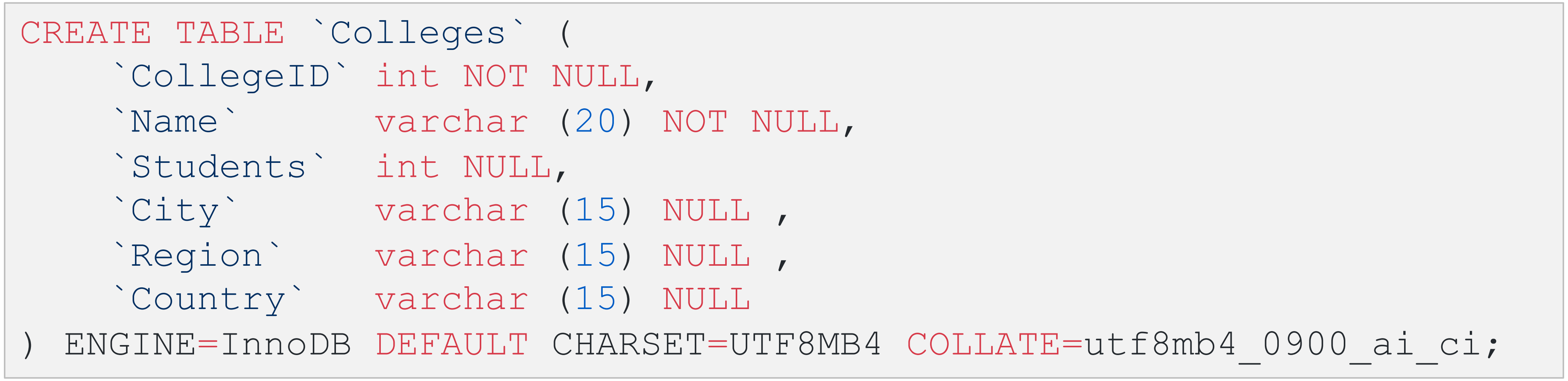
Creating tables

- Table name
- Table column names
- Datatype for each column
- Null status of each column – is null allowed or not
- Optional:
  - Default values
  - Column constraints – primary key, unique
  - Constraint reference checks

Create table

CREATE TABLE *TableName (ColumnList)*

[[**DEFAULT**] **CHARACTER SET** *CharsetName*]



```
CREATE TABLE `Colleges` (
    `CollegeID` int NOT NULL,
    `Name`      varchar (20) NOT NULL,
    `Students`  int NULL,
    `City`      varchar (15) NULL ,
    `Region`    varchar (15) NULL ,
    `Country`   varchar (15) NULL
) ENGINE=InnoDB DEFAULT CHARSET=UTF8MB4 COLLATE=utf8mb4_0900_ai_ci;
```

Drop table

DROP TABLE *TableName*

```
DROP TABLE IF EXISTS Colleges;
```

Add primary keys

```
DROP TABLE IF EXISTS Colleges;

CREATE TABLE `Colleges` (
    `CollegeID` int NOT NULL AUTO_INCREMENT,
    `Name` varchar (20) NOT NULL,
    `Students` int NULL,
    `City` varchar (15) NULL ,
    `Region` varchar (15) NULL ,
    `Country` varchar (15) NULL ,
    PRIMARY KEY (`CollegeID`)
) ENGINE=InnoDB DEFAULT CHARSET=UTF8MB4 COLLATE=utf8mb4_0900_ai_ci;
```

Datatypes

Types on other DB platforms	**MySQL Type**
BOOL	TINYINT
BOOLEAN	TINYINT
CHARACTER VARYING(*M*)	VARCHAR(*M*)
FIXED	DECIMAL
FLOAT4	FLOAT
FLOAT8	DOUBLE
INT1	TINYINT
INT2	SMALLINT
INT3	MEDIUMINT
INT4	INT
INT8	BIGINT
LONG VARBINARY	MEDIUMBLOB
LONG VARCHAR	MEDIUMTEXT
LONG	MEDIUMTEXT
MIDDLEINT	MEDIUMINT
NUMERIC	DECIMAL

Character types are the best choice when you have leading zeros, like zip codes

Null

- Primary keys should never be null
- If the field can be unknown, allow nulls
- A foreign key may be unknown for some time
  - A book may be entered before the publisher is known

Table creation process

- Decide the datatype of each column
- Decide which columns can be null, which should not
- Decide which columns need to be unique
- Note primary-foreign key pairings

Table meta data

Column name	Datatype	Null?	Unique?	Keys
CollegeID	int	Not null	yes	primary
Name	varchar(20)	Not null	no	
Students	int	Null	no	

Indexes

- Indexes speed the retrieval of data
- Just like a book index lets you find pages faster

```
DROP TABLE IF EXISTS Colleges;

CREATE TABLE `Colleges` (
    `CollegeID` int NOT NULL AUTO_INCREMENT,
    `Name` varchar (20) NOT NULL,
    `Students` int NULL,
    `City` varchar (15) NULL ,
    `Region` varchar (15) NULL ,
    `Country` varchar (15) NULL ,
    PRIMARY KEY (`CollegeID`),
    INDEX `CollegeID` (`CollegeID` ASC),
    INDEX `Name` (`Name` ASC)
) ENGINE=InnoDB DEFAULT CHARSET=UTF8MB4 COLLATE=utf8mb4_0900_ai_ci;
```

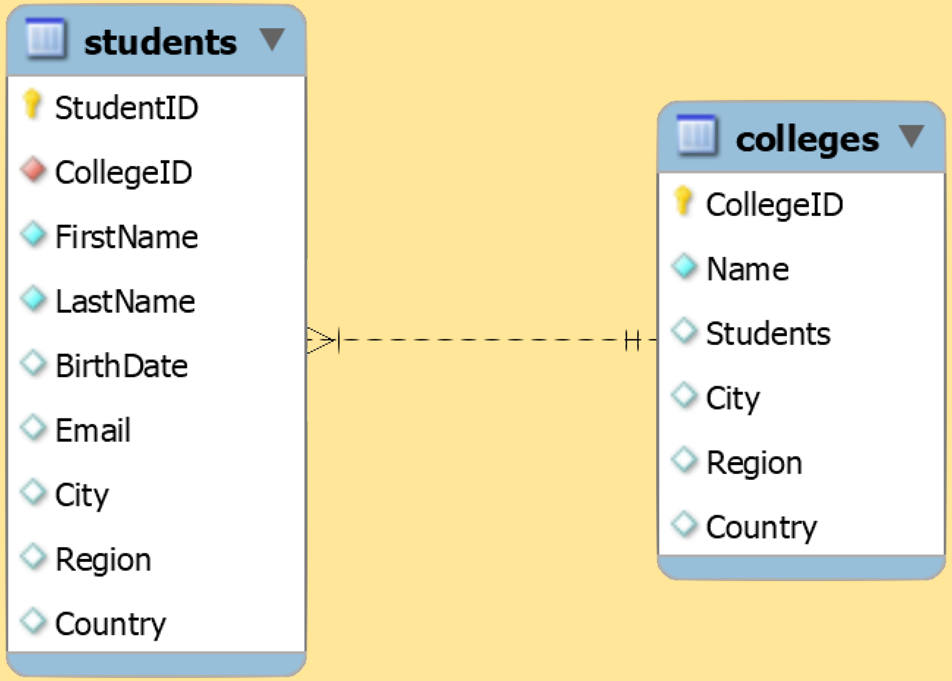
When to index

- Cost: Building and maintaining and index takes time and storage
- Cost: Inserting/deleting data takes a little longer
- Near worm (write once read many) is ideal

- Index
  - Heavily read columns
  - Primary keys – most of the time
  - Columns used in joins
  - Columns accessed in sorted order
  - Columns searched for ranges of values

Active Learning: add “students” table

- Add columns
- Add PKs
- Add FKs
- Add Indexes



students

StudentID
CollegeID
FirstName
LastName
BirthDate
Email
City
Region
Country

colleges

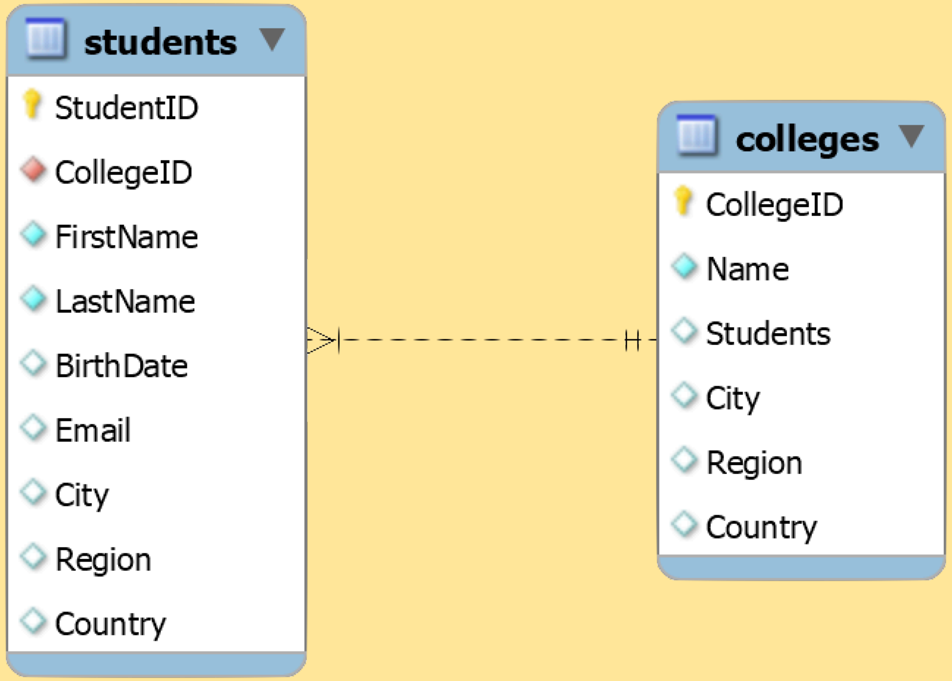
CollegeID
Name
Students
City
Region
Country

Adding new rows

INSERT INTO *TableName [(column1 [, column2] ...)]*

```
INSERT INTO `Colleges` VALUES(1,'MIT',11,'Cambridge','MA','USA');
INSERT INTO `Colleges` VALUES(2,'Brown',9,'Providence','RI','USA');
INSERT INTO `Colleges` VALUES(3,'Dartmouth',6,'Hanover','NH','USA');
```

Active Learning:
create input rows for students table



Updating rows

UPDATE *TableName* **SET** *ColumnName = value*

```
SET SQL_SAFE_UPDATES = 0;
UPDATE Colleges SET Country='U.S.';
```

Be careful, you will update all rows. Next we will limit with the WHERE clause.

* Useful when you add a new column to an existing DB and need to initialize

The where clause

UPDATE *TableName*

SET *ColumnName = value*

WHERE *whereCondition*

```
UPDATE Colleges
SET Country='USA'
WHERE Region='MA'
```

Removing rows

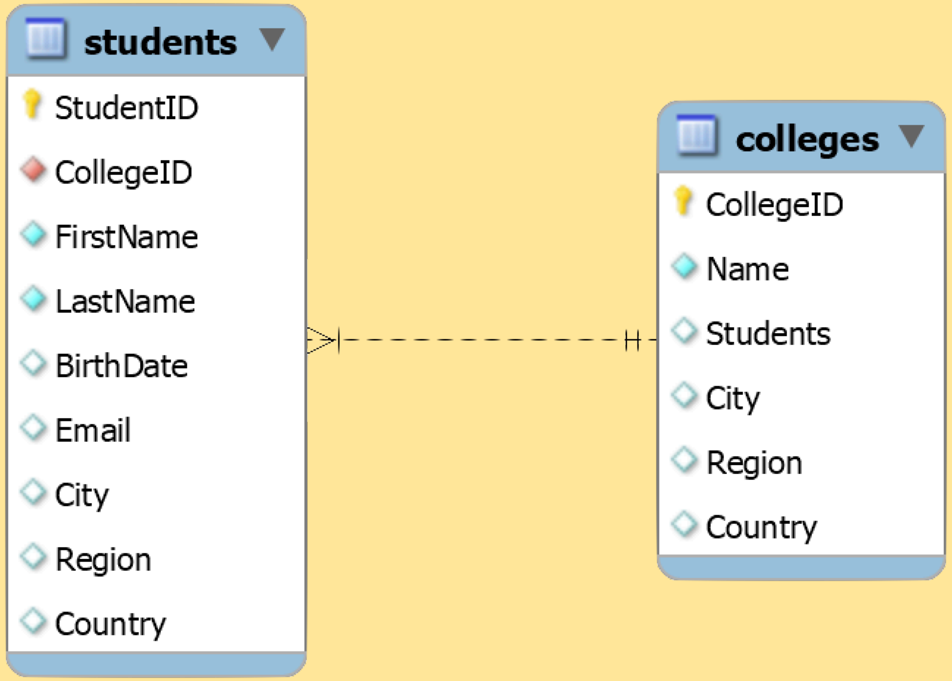
DELETE FROM *TableName*

WHERE *whereCondition*

```
DELETE FROM Colleges
WHERE Name='NYU'
```

Active Learning: create ”education” database installation script

- Create database
- Create tables
- Create input data for colleges
- Create input data for students


students
StudentID
CollegeID
FirstName
LastName
BirthDate
Email
City
Region
Country
colleges
CollegeID
Name
Students
City
Region
Country