Notation: The notation used here is a little different from the notation used in lectures.

|  |  |  |
| --- | --- | --- |
| Notation | Lecture notes | MySQLWorkbench |
| Optional relationship | Dotted line | Dotted line is an circle at the end |
| Mandatory relationship | Solid line | Dotted line with vertical line at the end. |
| The ‘many’ side of a relationship | Crows foot | Crows foot |
| The ‘one’ side of a relationship | Single line | Single line with a vertical line at the end |

MySQL Workbench *symbols* and their meaning:

|  |  |
| --- | --- |
| Symbol | Meaning |
|  | The ‘one’ side of a mandatory relationship |
|  | The many side of a mandatory relationship |
|  | The ‘one’ side of an optional relationship |
|  | The many side of an optional relationship |

*SQL Statements that you will need to cover for the Class Tests:*

*Show data*

SELECT \* FROM emp;

SELECT DISTINCT example FROM example;

SELECT UNIQUE example FROM example;

*Show data for condition*

SELECT FROM example WHERE condition;

*Show data for 2 and conditions*

SELECT FROM example WHERE condition AND condition;

*Show data for condition between 2 values*

SELECT FROM example WHERE condition BETWEEN condition AND condition;

*Show if data is in a set*

SELECT FROM example WHERE value IN (1,2,3);

SELECT FROM example WHERE value NOT IN (1,2,3);

*Show max, min, avgand count*

SELECT max(example);

SELECT min(example);

SELECT avg(example);

SELECT count(example);

SELECT count(\*) example from example;

*Create column for data*

SELECT ename AS ‘Employee Name’, sal\*12 AS ‘Total Salary’ FROM emp;

SELECT example   
FROM example  
WHERE example LIKE ‘b%’; *// names beginning with b*

SELECT example

FROM example

WHERE example LIKE ‘\_anet’; *//..unknown first letter*

SELECT example

FROM example

WHERE example LIKE ‘%S’; *//ending with the letter s*

*Lecture 3: Example of joining and grouping*

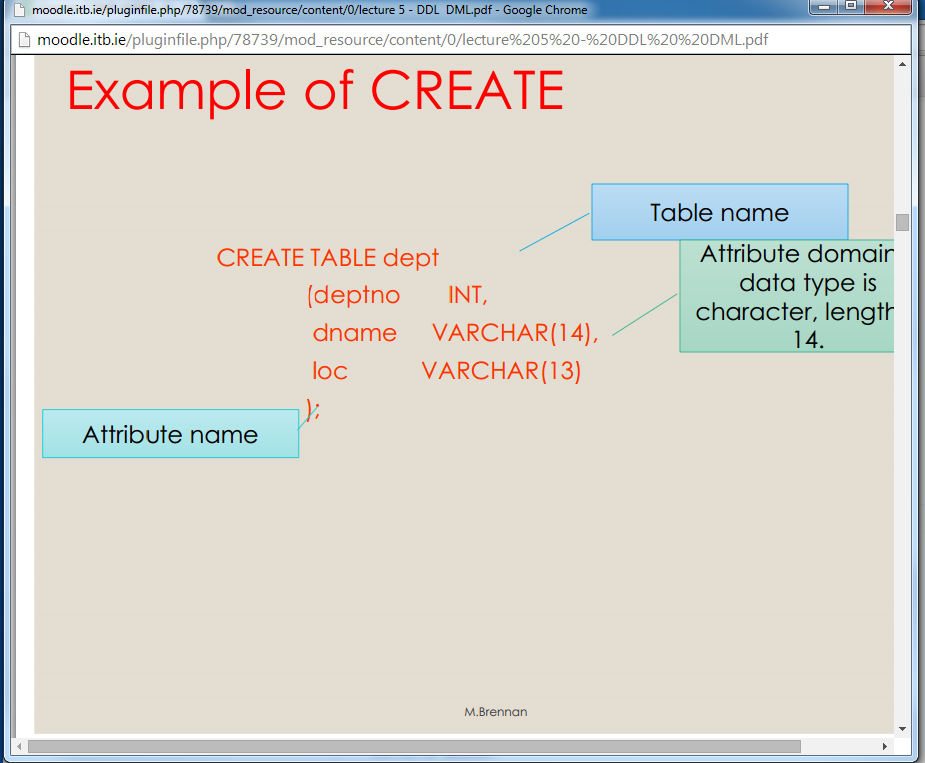
SELECT emp.ename, emp.sal, dept.deptno

FROM emp inner join dept

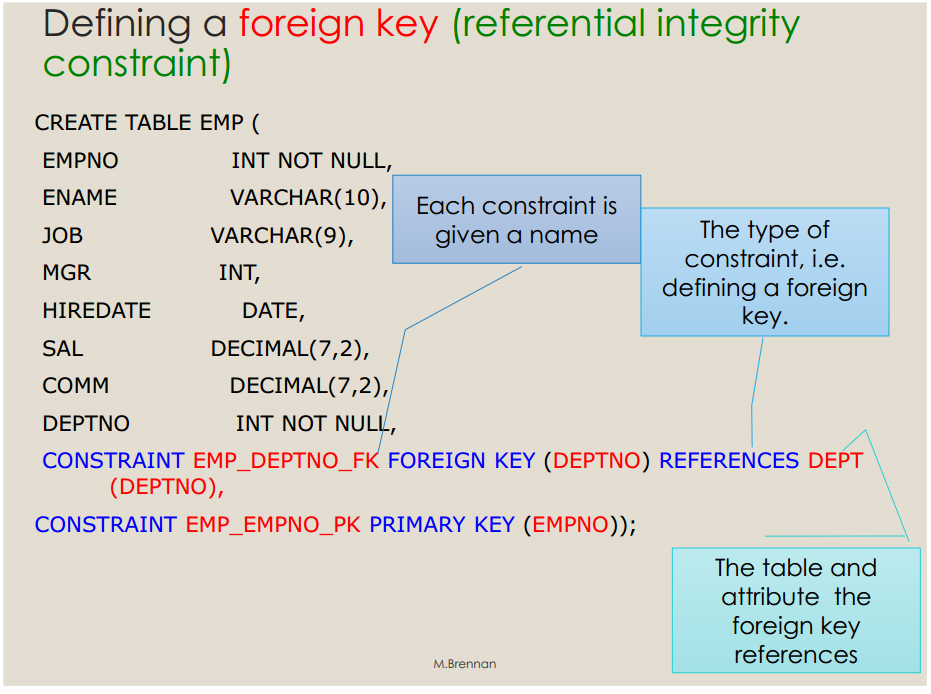
WHERE condition

GROUP BY dept.deptno;

*Used to create table*



*Used to create table and select pk and fk*



*Used to add primary key*

ALTER TABLE dept

ADD CONSTRAINT dept\_loc\_fk FOREIGN KEY loc REFERENCES location (loc);

*Used to disable pk*

ALTER TABLE emp

DISABLE CONSTRAINT emp\_pk CASCADE;

*Used to drop column*

ALTER TABLE dept

DROP COLUMN job;

*Used to edit created columns*

ALTER TABLE dept

MODIFY dname VARCHAR(20) NOT NULL;

*Used to add columns*

ALTER TABLE dept

ADD location VARCHAR(20);

*Adds data to the column*

INSERT INTO dept

VALUES (05, ‘development’, ‘Dublin’);

*Edits inputted values based on a condition to locate correct row*

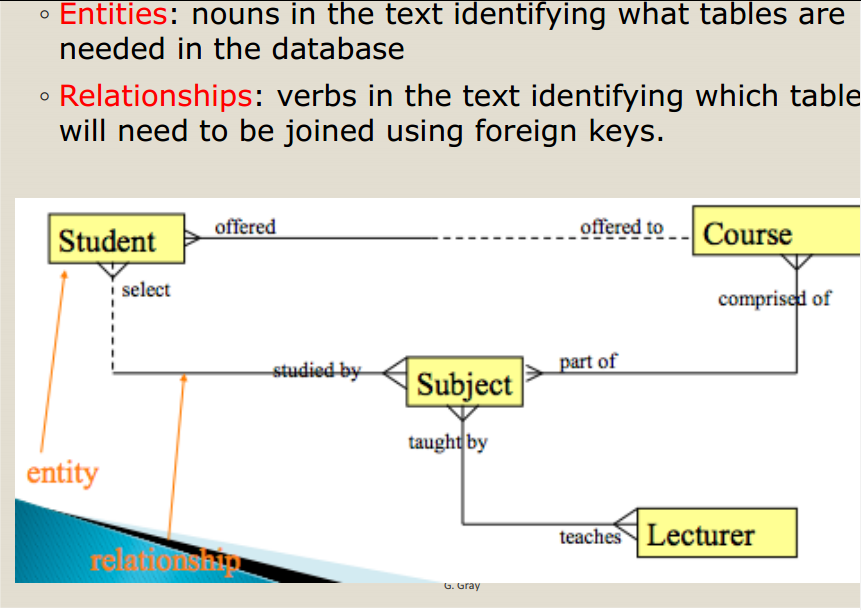
UPDATE table\_name

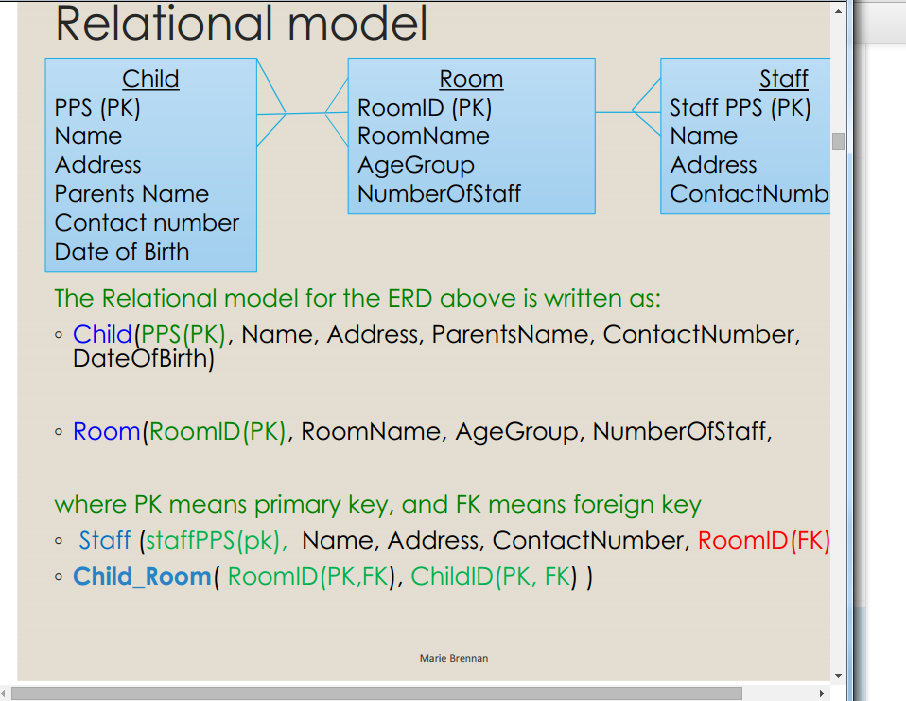
SET column\_1 = new\_value,

column\_2 = new\_value, ….

[WHERE user number = 1]

*Entities*





Fk goes in either if 1:1, many in 1: m, and the new table created in an M: N.

*Example normal form*

◦ Student (Student ID(PK), student name, lecturer

name, course description)

◦ Does a student ID identify a specific student’s name?

◦ Does a student ID identify a specific lecturer’s name?

◦ Does a student ID identify a specific course

description?

*Only student name is functionally dependent on*

*Student ID. The other attributes are in the wrong*

*table.*

Example 2

◦ Student\_Subject(Student ID(PK), subject ID(FK),

grade, student name, subject name)

◦ Do you need both the student ID and the subject ID

to find the grade a student got in a particular subject?

◦ Do you need both the student ID and the subject ID

to get a student’s name?

◦ Do you need both the student ID and the subject ID

to get the subject’s name?

*Only grade is functionally dependent on Student ID*

*AND subject ID. The other attributes are in the wrong*

*table.*

**Student ID(PK), student name, subject name, grade**

**99123456 Kelly Databases C**

**99123456 Kelly Software Dev B**

**99123456 Kelly Networking C+**

Example 3

• Does a student ID identify a specific student’s name?

• Does a student ID identify a specific subject’s name?

• Does a student ID identify a specific grade?

*Only student name is functionally dependent on Student*

*ID. The other attributes are in the wrong table.*

Recap – Functionally dependent means the attribute has one unique value that

can be determined from the key field.

There are three ways in which an attribute is NOT functionally dependent on the

primary key, as illustrated in the three examples done previously.

• Identifying these scenarios is done by following the three steps of Normalization:

1. Bring to 1st normal form – remove repeating groups, i.e Example 3 above.

2. Bring to 2nd normal form – remove partial dependencies, i.e. Example 2 above

3. Bring to 3rd normal form – remove transitive dependencies, i.e. Example 1 above

•

Once in third normal form (3NF), the tables

are well-structured

