CS7025 Programming for Digital Media

Lesson 11 – MySQL

Recap

SQL

Relational databases to organise data in a structured way

Interact with the database using SQL

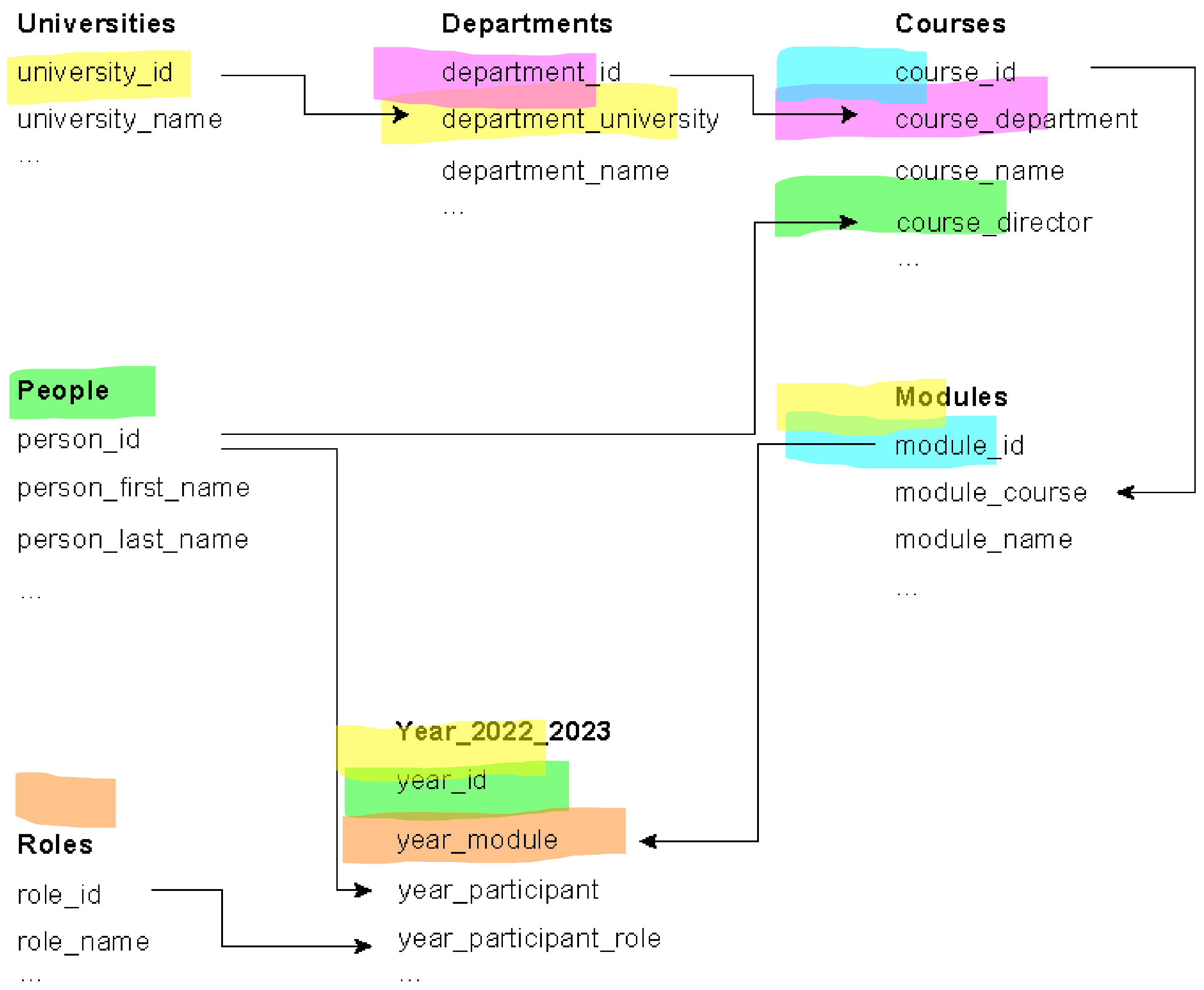
Last week we looked at simple queries (**CRUD**)

```
Create     // INSERT
Read       //  SELECT
Update     // UPDATE
Delete     // DELETE
```

Trinity College Dublin
Coláiste na Tríonóide, Baile Átha Cliath
The University of Dublin

Databases

Relational Databases



Databases

Relational Databases

module_id	module_name	ECTS
CS7025	Programming for Digital Media	10
CS7026	Authoring for Digital Media	10
CS7027	Contextual Media	10
CS7028	Audio, Video and Sensor Technologies	10

role_id	role_name
1	Lecturer
2	Demonstrator
3	Student
…	…

year_id	year_module	year_participant	year_participant_role
2389	CS7025	237	1
2390	CS7025	278	2
2391	CS7025	299	2
2392	CS7025	310	3

person_id	person_first_name	person_last_name
237	Joris	Vreeke
278	Rose	Connolly
299	Hassan	Zaal
310	John	Doe

MySQL

There are 2 types of tables:

1. define or describe entities
   Think of a Stadium, or a Team or Players >
   team_id, team_name
2. use the data, management tables
   Think of match data >
   match_id > home_team, away_team

- The second one is used to tie tables of the first type together

MySQL

JOINS

To make the data more legible you have to connect the tables by using the JOIN command

```
SELECT *
FROM matches
(INNER) JOIN teams ON match_home_team = team_id;
```

Note: you can tie multiple tables together by using multiple JOINS in one SQL command

MySQL
JOINS

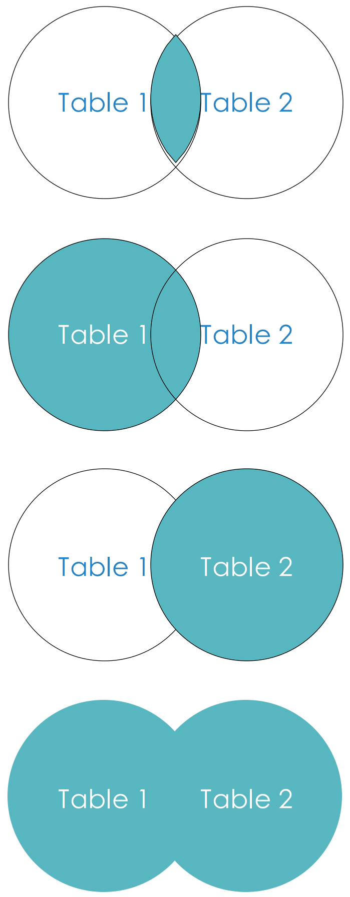
There are 4 types of JOINS

(INNER) JOIN: Returns records that have matching values in both tables

LEFT (OUTER) JOIN: Returns all records from the left table, and the matched records from the right table

RIGHT (OUTER) JOIN: Returns all records from the right table, and the matched records from the left table

FULL (OUTER) JOIN: Returns all records when there is a match in either left or right table



MySQL
LIKE

Instead of using an id/key to filter results you can look for entries by using the LIKE keyword

It uses wildcards %

```
SELECT *
FROM players
JOIN teams ON player_team=team_id
WHERE team_name LIKE "%relan%"
```

MySQL
FUNCTIONS

SQL has built in functions for getting the total number of something (see below) or its average or the minimum or maximum value of a dataset:

```
SELECT COUNT(player_id), team_country
FROM players
JOIN teams ON player_team = team_id
GROUP BY player_team
ORDER BY COUNT(player_id) DESC, team_country;
```

Try it yourself

Scratch

Thank You

Trinity College Dublin
Coláiste na Tríonóide, Baile Átha Cliath
The University of Dublin