Managing Data & Databases

Session 10
Let the links flourish

Today's Dose of SQL

- DML
  - INNER JOIN
  - LEFT OUTER JOIN
  - RIGHT OUTER JOIN
  - FULL OUTER JOIN
  - UNION ALL
  - UNION DISTINCT

UNION ALL vs. UNION DISTINCT

- Simple Explanation
  - UNION DISTINCT = UNION = DISTINCT(UNION ALL)
- Complex Explanation
  - UNION merges two tables vertically and removes duplicate entries. When duplicate entries are required as a part of the query response, UNION ALL must be used.
- Example
  - SELECT 'Text Field' UNION SELECT 'Text Field'; =>

Text Field

  - SELECT 'Text Field' UNION ALL SELECT 'Text Field'; =>

Text Field
Text Field

JOIN vs. RIGHT JOIN vs. LEFT JOIN

- Explanation
  - JOIN merges two tables vertically and returns the records in which the two tables have a common value in a certain field.
  - LEFT JOIN returns the same results as JOIN in addition to all the records in the left table that have no corresponding record in the right table.
  - RIGHT JOIN returns the same results as JOIN in addition to all the records in the right table that have no corresponding record in the left table.
- Example
  - SELECT people.id, first_name, last_name, label AS team_label
    FROM people JOIN team
    ON team=team.id;
  - SELECT people.id, first_name, last_name, label AS team_label
    FROM people LEFT JOIN team
    ON team=team.id;
  - SELECT people.id, first_name, last_name, label AS team_label
    FROM people RIGHT JOIN team
    ON team=team.id;

What does a join look like?

id	first_name	last_name	team
1	Mahmood	Zargar	NULL
11	Omar	Abdelkader	1
21	Saharea	Ahamed	1
31	Liam	Amilhastre	2
41	Shehryar	Bajwa	2
51	Emily	Barber	2
61	Laura	Easton	3
71	Joshua	Frank	3
81	Matthew	Fryml	3
91	Xavier	Guérette	1
101	Blaire	Hinton	NULL
111	Yuan Ping	Jin	NULL

JOIN

id	label
1	Team One
2	Team Two
3	Team Three
4	Team four

=

id	first_name	last_name	label
11	Omar	Abdelkader	Team One
21	Saharea	Ahamed	Team One
91	Xavier	Guérette	Team One
31	Liam	Amilhastre	Team Two
41	Shehryar	Bajwa	Team Two
51	Emily	Barber	Team Two
61	Laura	Easton	Team Three
71	Joshua	Frank	Team Three
81	Matthew	Fryml	Team Three

What does a right join look like?

id	first_name	last_name	team
1	Mahmood	Zargar	NULL
11	Omar	Abdelkader	1
21	Saharea	Ahamed	1
31	Liam	Amilhastre	2
41	Shehryar	Bajwa	2
51	Emily	Barber	2
61	Laura	Easton	3
71	Joshua	Frank	3
81	Matthew	Fryml	3
91	Xavier	Guérette	1
101	Blaire	Hinton	NULL
111	Yuan Ping	Jin	NULL

RIGHT JOIN

id	label
1	Team One
2	Team Two
3	Team Three
4	Team four

=

id	first_name	last_name	label
11	Omar	Abdelkader	Team One
21	Saharea	Ahamed	Team One
91	Xavier	Guérette	Team One
31	Liam	Amilhastre	Team Two
41	Shehryar	Bajwa	Team Two
51	Emily	Barber	Team Two
61	Laura	Easton	Team Three
71	Joshua	Frank	Team Three
81	Matthew	Fryml	Team Three
NULL	NULL	NULL	Team four

What does a left join look like?

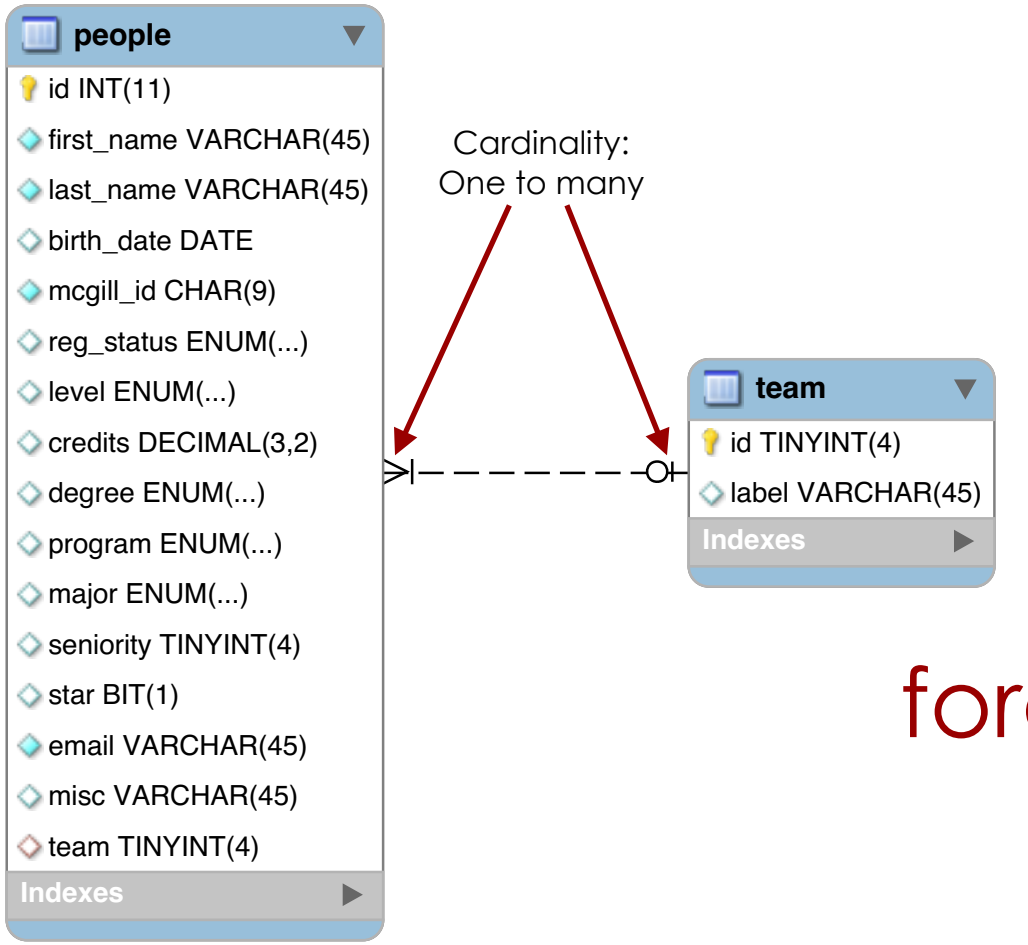
id	first_name	last_name	team
1	Mahmood	Zargar	NULL
11	Omar	Abdelkader	1
21	Saharea	Ahamed	1
31	Liam	Amilhastre	2
41	Shehryar	Bajwa	2
51	Emily	Barber	2
61	Laura	Easton	3
71	Joshua	Frank	3
81	Matthew	Fryml	3
91	Xavier	Guérette	1
101	Blaire	Hinton	NULL
111	Yuan Ping	Jin	NULL

LEFT JOIN

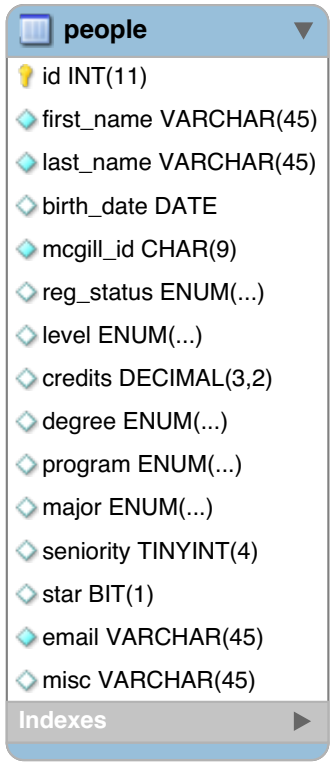
id	label
1	Team One
2	Team Two
3	Team Three
4	Team four

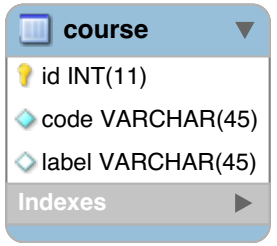
=

id	first_name	last_name	label
1	Mahmood	Zargar	NULL
11	Omar	Abdelkader	Team One
21	Saharea	Ahamed	Team One
31	Liam	Amilhastre	Team Two
41	Shehryar	Bajwa	Team Two
51	Emily	Barber	Team Two
61	Laura	Easton	Team Three
71	Joshua	Frank	Team Three
81	Matthew	Fryml	Team Three
91	Xavier	Guérette	Team One
101	Blaire	Hinton	NULL
111	Yuan Ping	Jin	NULL

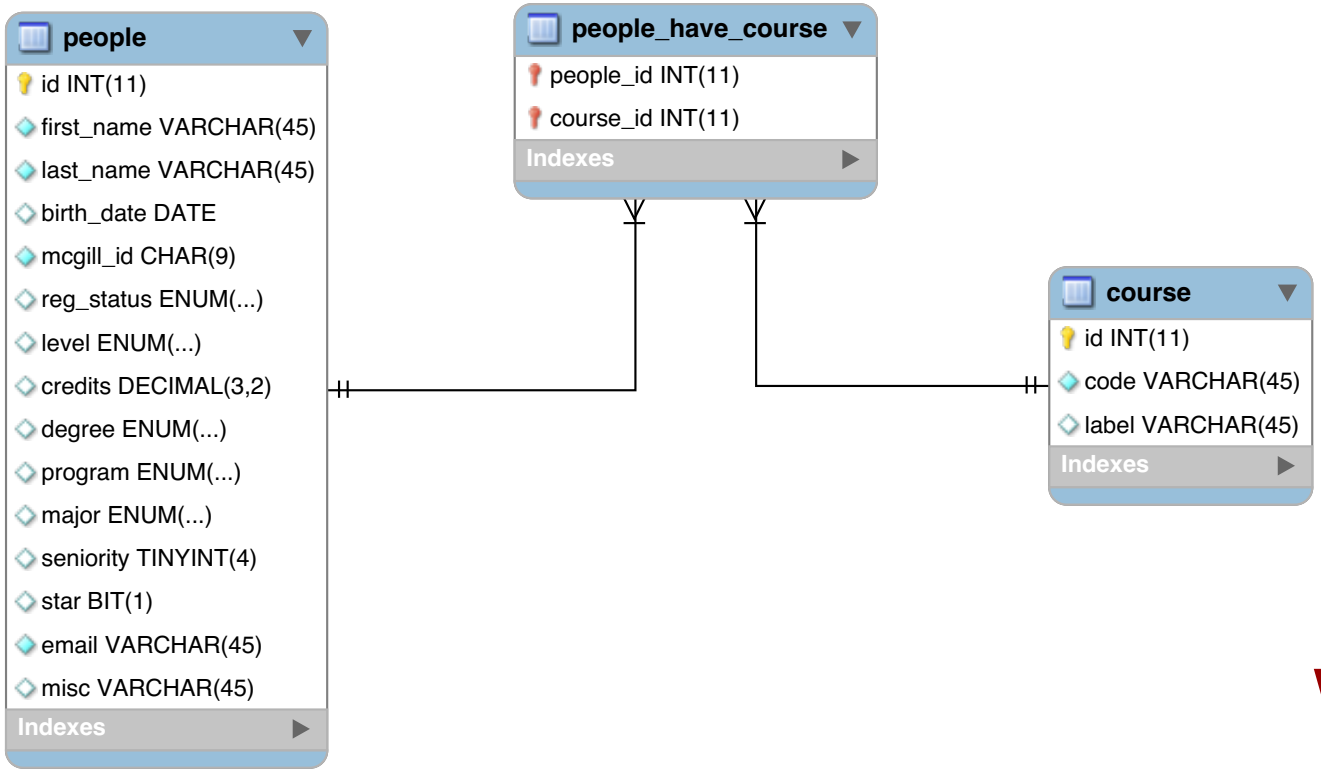


What does a foreign key look like?

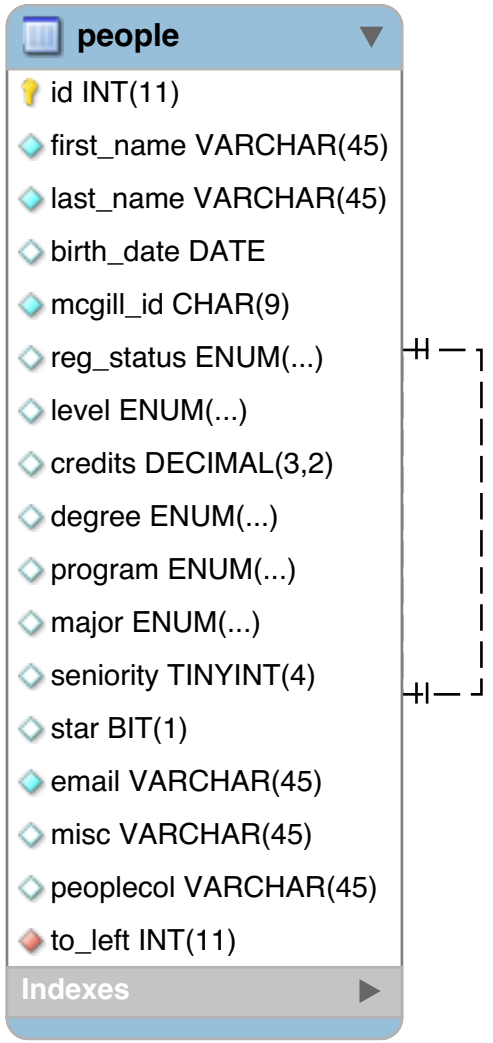




What about many-to-many relationships?



What about many-to-many relationships?



What about nested/recursive relationships?