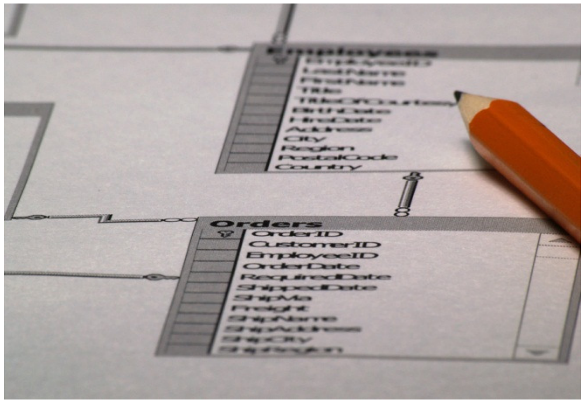


EducationAnytime LLC

Course

MySQL Foundations – Querying Data

Truth Tables

OR

A	B	Result
FALSE	FALSE	FALSE
TRUE	FALSE	TRUE
FALSE	TRUE	TRUE
TRUE	TRUE	TRUE

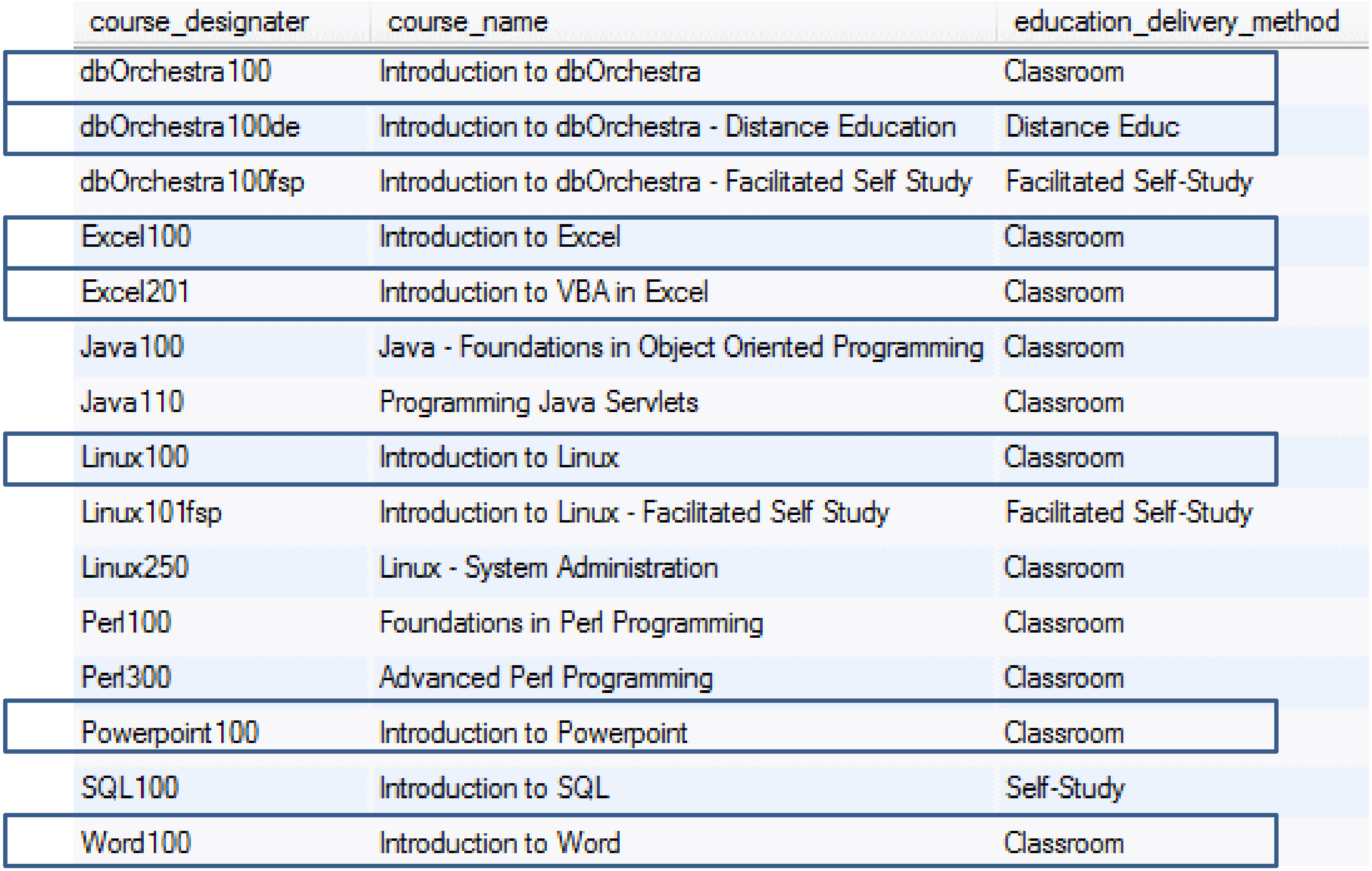
If any condition is TRUE the result is TRUE

AND

A	B	Result
FALSE	FALSE	FALSE
TRUE	FALSE	FALSE
FALSE	TRUE	FALSE
TRUE	TRUE	TRUE

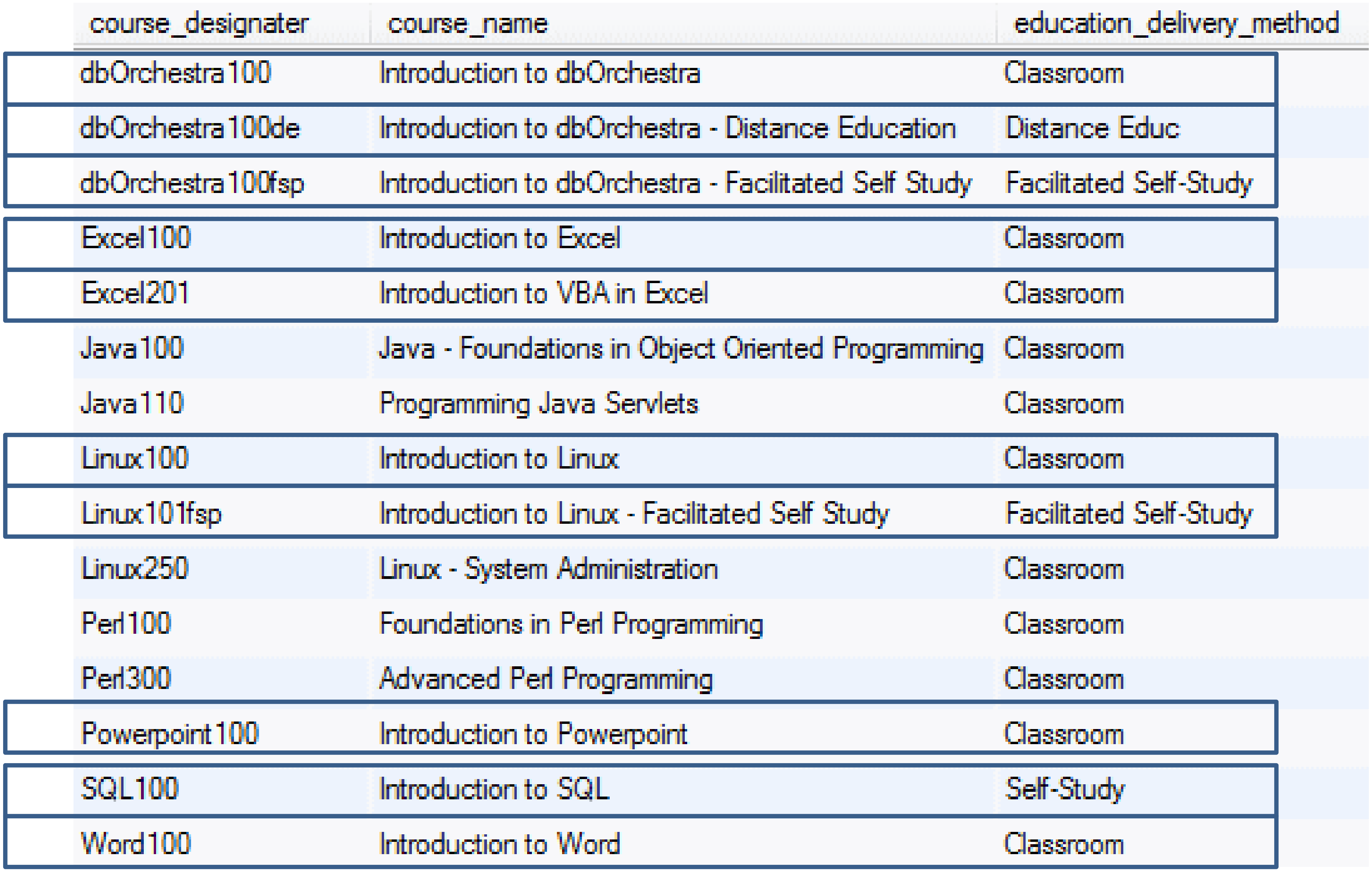
All conditions must be TRUE for the result to be TRUE

```
WHERE
    education_delivery_method = 'Classroom' AND course_name LIKE 'Intro%'
    OR
    education_delivery_method = 'Distance Educ' AND course_name LIKE 'Intro%'
```



course_designater	course_name	education_delivery_method
dbOrchestra100	Introduction to dbOrchestra	Classroom
dbOrchestra100de	Introduction to dbOrchestra - Distance Education	Distance Educ
dbOrchestra100fsp	Introduction to dbOrchestra - Facilitated Self Study	Facilitated Self-Study
Excel100	Introduction to Excel	Classroom
Excel201	Introduction to VBA in Excel	Classroom
Java100	Java - Foundations in Object Oriented Programming	Classroom
Java110	Programming Java Servlets	Classroom
Linux100	Introduction to Linux	Classroom
Linux101fsp	Introduction to Linux - Facilitated Self Study	Facilitated Self-Study
Linux250	Linux - System Administration	Classroom
Perl100	Foundations in Perl Programming	Classroom
Perl300	Advanced Perl Programming	Classroom
Powerpoint100	Introduction to Powerpoint	Classroom
SQL100	Introduction to SQL	Self-Study
Word100	Introduction to Word	Classroom

```
WHERE
   education_delivery_method = 'Classroom' AND
   education_delivery_method = 'Distance Educ'
   OR   course_name LIKE 'Intro%'
```



course_designater	course_name	education_delivery_method
dbOrchestra100	Introduction to dbOrchestra	Classroom
dbOrchestra100de	Introduction to dbOrchestra - Distance Education	Distance Educ
dbOrchestra100fsp	Introduction to dbOrchestra - Facilitated Self Study	Facilitated Self-Study
Excel100	Introduction to Excel	Classroom
Excel201	Introduction to VBA in Excel	Classroom
Java100	Java - Foundations in Object Oriented Programming	Classroom
Java110	Programming Java Servlets	Classroom
Linux100	Introduction to Linux	Classroom
Linux101fsp	Introduction to Linux - Facilitated Self Study	Facilitated Self-Study
Linux250	Linux - System Administration	Classroom
Perl100	Foundations in Perl Programming	Classroom
Perl300	Advanced Perl Programming	Classroom
Powerpoint100	Introduction to Powerpoint	Classroom
SQL100	Introduction to SQL	Self-Study
Word100	Introduction to Word	Classroom

Join Types

Person	
lastname	person_pk
Smith	1
Jones	2
Jonhnson	3
Vilander	4
Jones	5
Darcy	6
Melrose	7
Atkins	8

phone		
person_fk	phn_typ	number
1	C	555-1234
2	W	555-3422
2	C	555-9800
3	C	555-8777
4	W	444-2323
7	W	555-1221
4	C	555-2222

- Join

```
SELECT lastname, phn_typ, number
FROM Person
JOIN phone ON
     phone.person_fk = Person.person_pk
WHERE person_pk >=4
```

lastname	phn_type	number
Vilander	W	444-2323
Vilander	C	555-2222
Melrose	W	555-1221

- Left Join

```
SELECT lastname, phn_typ, number
FROM Person
LEFT JOIN phone ON
     phone.person_fk = Person.person_pk
WHERE person_pk >= 4
```

lastname	phn_type	number
Vilander	W	444-2323
Vilander	C	555-2222
Jones		
Darcy		
Melrose	W	555-1221
Atkins		

Join Types

- Syntax is straight forward.
- Use join syntax instead of joining in the Where clause.

```
SELECT lastname, phn_typ, number
FROM Person
JOIN phone ON
     phone.person_fk = Person.person_pk
WHERE person_pk >=4
```

Instead of

```
SELECT lastname, phn_typ, number
FROM Person, phone
WHERE person_pk >=4 AND
      phone.person_fk = Person.person_pk
```

- How do I know what tables to join?

DRI (Declarative Referential Integrity)

E-R Diagrams

Purpose of E-R Diagrams

E-R Relationships

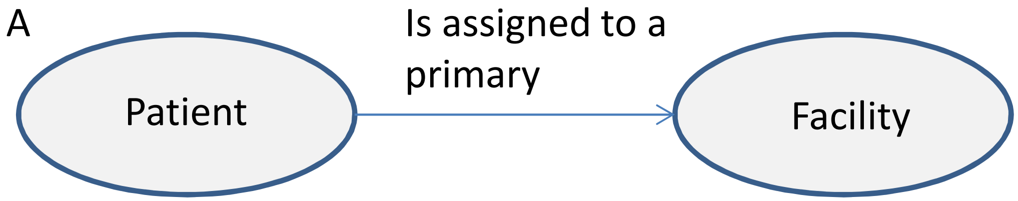
- One to One (1-1)
- One to Many (1-M)
- Many to Many (M-M)

E-R Diagrams

- To determine table cardinality

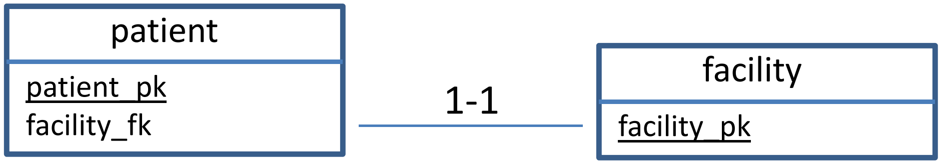
Entity Relationships – One to One (1-1)

Logical



A Patient Is assigned to a primary Facility

Physical



patient

patient_pk
facility_fk

1-1

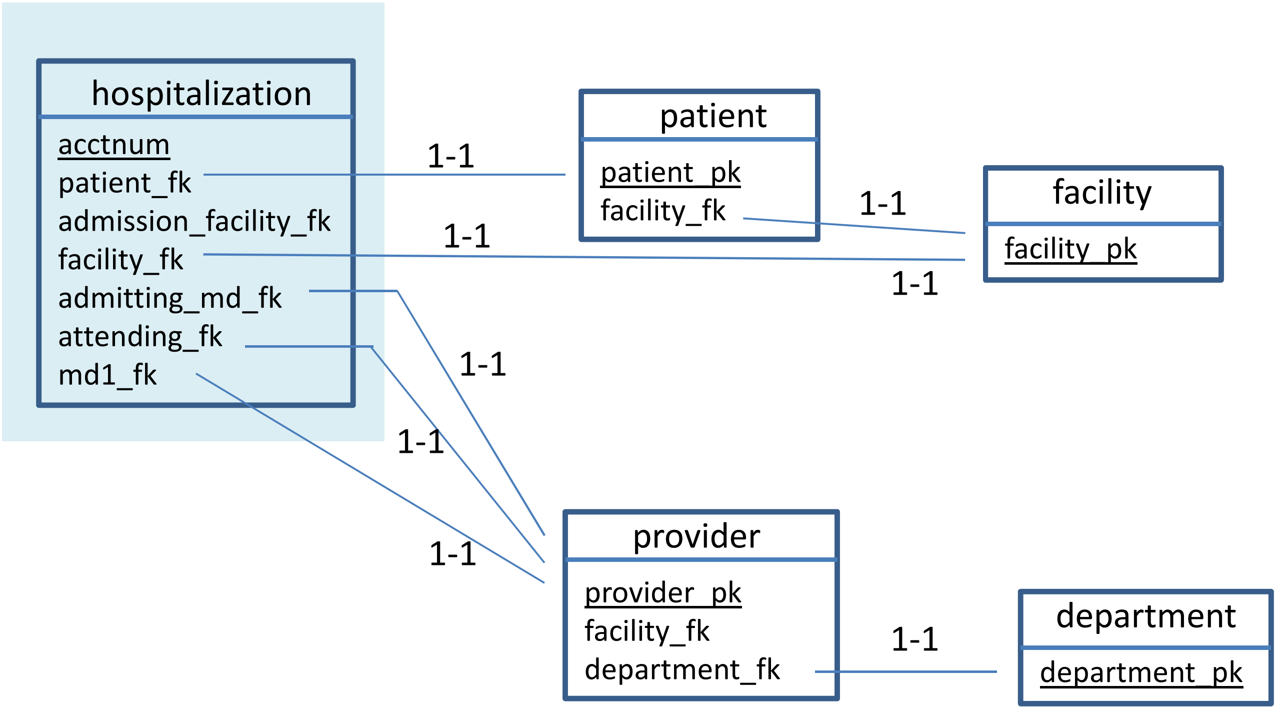
facility

facility_pk

How do I know that the join is 1 to 1?

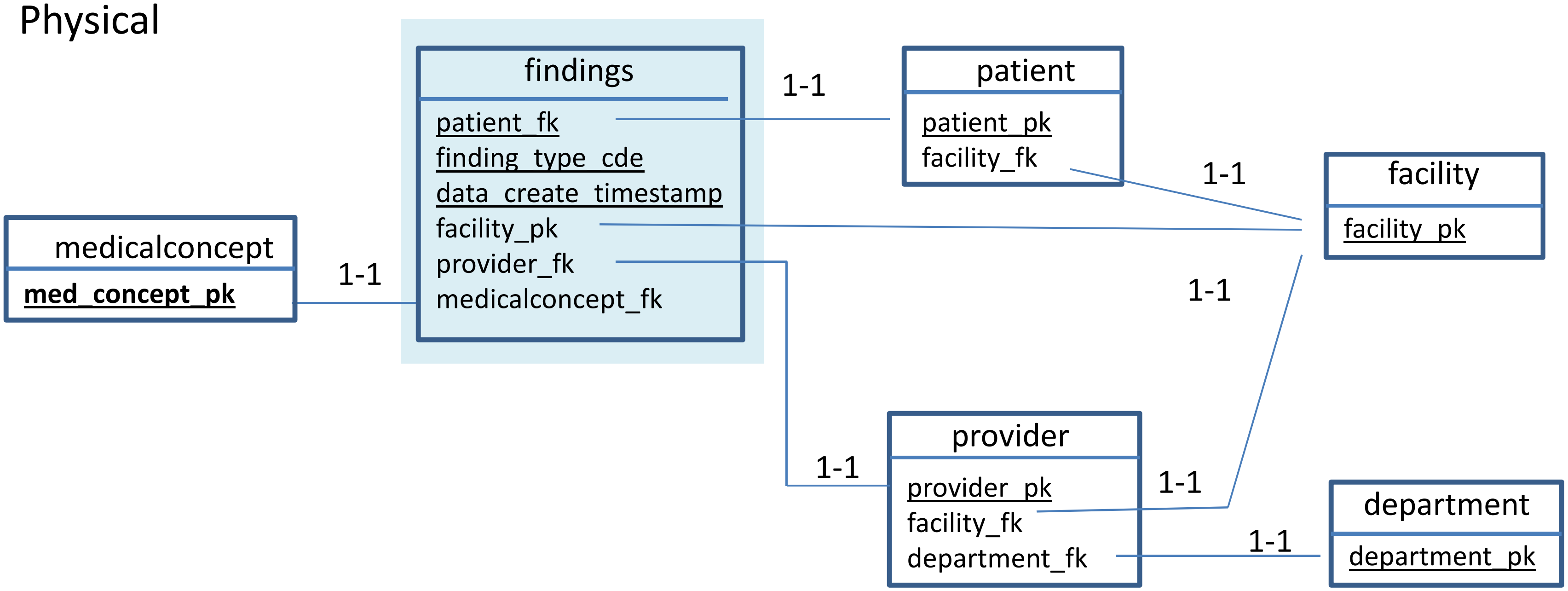
Entity Relationships Diagrams - Healthcare Database – Hospitalization Perspective

Physical



Entity Relationships Diagrams - Healthcare Database – Findings Perspective

Physical



findings

patient_fk
finding_type_cde
data_create_timestamp
facility_pk
provider_fk
medicalconcept_fk

patient

patient_pk
facility_fk

facility

facility_pk

medicalconcept

med_concept_pk

provider

provider_pk
facility_fk
department_fk

department

department_pk

1-1

1-1

1-1

1-1

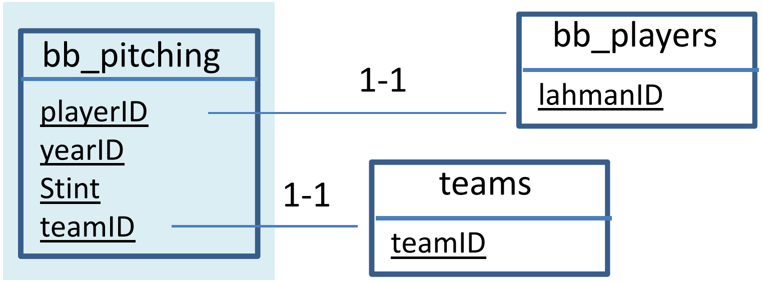
1-1

1-1

1-1

Entity Relationships Diagrams - trainwarehouse Database – Pitcher/Player Tables

Physical



bb_pitching

playerID
yearID
Stint
teamID

1-1

bb_players

lahmanID

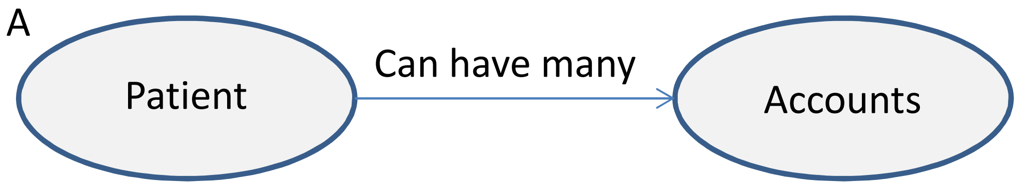
1-1

teams

teamID

Entity Relationships – One to Many (1-M)

Logical



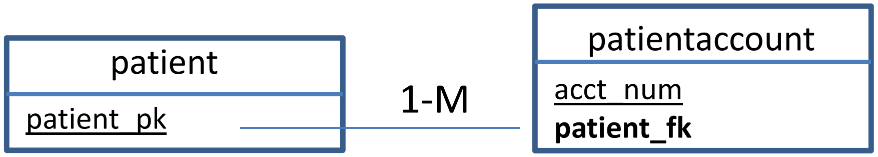
A

Patient

Can have many

Accounts

Physical



patient

patient_pk

1-M

patientaccount

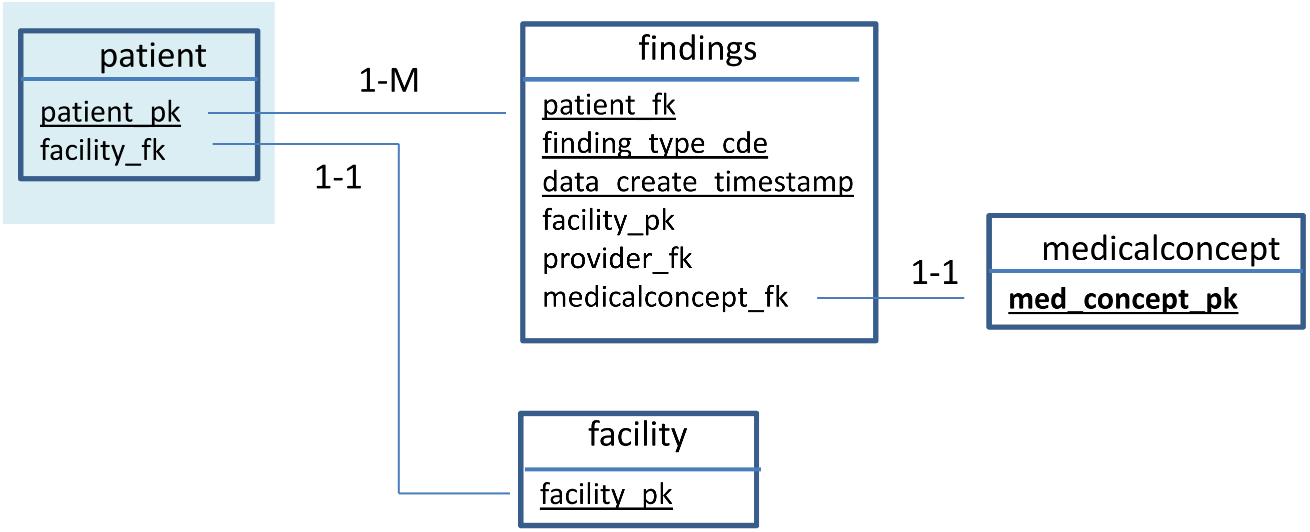
acct_num

patient_fk

How do I know that the join is 1 to M?

Entity Relationships Diagrams - Healthcare Database – Patient Perspective To Findings

Physical



patient

patient_pk
facility_fk

1-M

1-1

findings

patient_fk
finding_type_cde
data_create_timestamp
facility_pk
provider_fk
medicalconcept_fk

1-1

medicalconcept

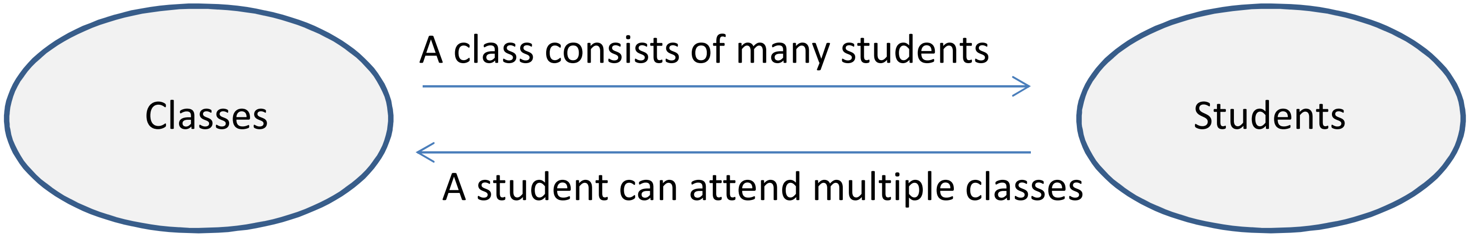
med_concept_pk

facility

facility_pk

Entity Relationships – Many to Many (M-M)

Logical



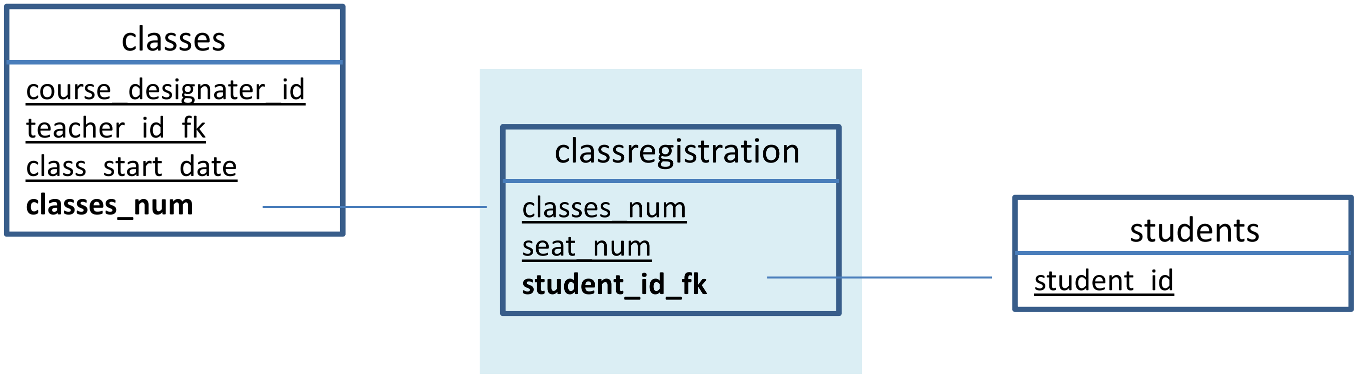
Classes

A class consists of many students

A student can attend multiple classes

Students

Physical



classes

course_designater_id
teacher_id_fk
class_start_date
classes_num

classregistration

classes_num
seat_num
student_id_fk

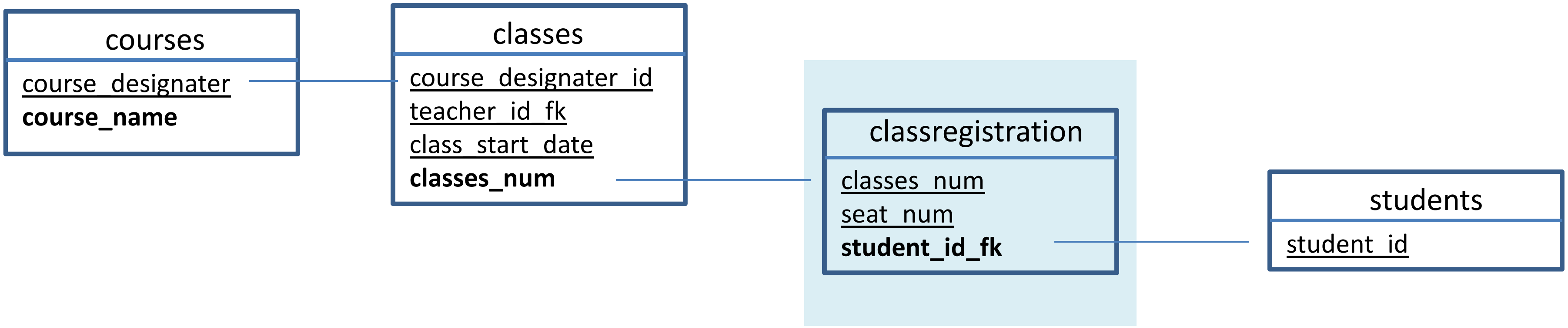
students

student_id

How do I know that the join is M to M?

Entity Relationships – Many to Many (M-M)

Physical



courses

course_designater

course_name

classes

course_designater_id

teacher_id_fk

class_start_date

classes_num

classregistration

classes_num

seat_num

student_id_fk

students

student_id