Relational Database Model

Characteristics of a Relational Table

- Perceived as a 2D structure of rows and columns
- Each row is a single entity occurrence within the entity set
- Each column is an attribute and has a distinct name
- Each intersection of a row and column is a single data value
- All values in a column conform to the same data format
- Each column has an attribute domain (specific range of values)
- Order of columns and rows is immaterial to the DBMS
- Each table must have a key that uniquely identifies each row

Characteristics of a Relational Table

- Table: Customer
- Attributes
  - Code (key, numeric)
  - Last Name (character string)
  - First Name (character string)
  - Initial (character)
  - Renew Date (date)
  - Agent Code (numeric)

CUS_CODE	CUS_LNAME	CUS_FNAME	CUS_INITIAL	CUS_RENEW_DATE	AGENT_CODE
10010	Ramas	Alfred	A	05-Apr-2018	502
10011	Dunne	Leona	K	16-Jun-2018	501
10012	Smith	Kathy	W	29-Jan-2019	502
10013	Olowski	Paul	F	14-Oct-2018	
10014	Orlando	Myron		28-Dec-2018	501
10015	O'Brian	Amy	B	22-Sep-2018	503
10016	Brown	James	G	25-Mar-2019	502
10017	Williams	George		17-Jul-2018	503
10018	Farriss	Anne	G	03-Dec-2018	501
10019	Smith	Olette	K	14-Mar-2019	503

- Note the empty cells—these are null values, i.e., unknown or no value

Keys

- Key – one or more attributes that determine other attributes
  - Key attribute – an attribute that is a part of the key
  - Composite key – key composed of multiple attributes
  - Ensures each row is uniquely identifiable
  - Ensures the integrity of the data
- Superkey – any set of attributes that can uniquely identify a row
- Candidate key – minimal superkey (no more attributes than necessary)
- Primary Key (PK) – selected candidate key to uniquely identify a row
- Secondary key – key used strictly for data retrieval purposes
- Surrogate key – used to simplify the identification of a row (section 5-3d)
- Foreign key (FK) – PK of a table placed in another to create a relationship

Keys

- Superkeys
  - CODE
  - CODE + LNAME
  - CODE + FNAME
  - CODE + INITIAL
  - CODE + LNAME + FNAME
  - CODE + RENEW DATE
  - CODE + all attributes
- In other words, any combination of attributes that contain key attributes

Table name: CUSTOMER **Database name: Ch03_InsureCo**
Primary key: CUS_CODE
Foreign key: AGENT_CODE

CUS_CODE	CUS_LNAME	CUS_FNAME	CUS_INITIAL	CUS_RENEW_DATE	AGENT_CODE
10010	Ramas	Alfred	A	05-Apr-2018	502
10011	Dunne	Leona	K	16-Jun-2018	501
10012	Smith	Kathy	W	29-Jan-2019	502
10013	Olowski	Paul	F	14-Oct-2018	
10014	Orlando	Myron		28-Dec-2018	501
10015	O'Brian	Amy	B	22-Sep-2018	503
10016	Brown	James	G	25-Mar-2019	502
10017	Williams	George		17-Jul-2018	503
10018	Farriss	Anne	G	03-Dec-2018	501
10019	Smith	Olette	K	14-Mar-2019	503

Table name: AGENT (only five selected fields are shown)
Primary key: AGENT_CODE
Foreign key: none

AGENT_CODE	AGENT_AREACODE	AGENT_PHONE	AGENT_LNAME	AGENT_YTD_SLS
501	713	228-1249	Alby	132735.75
502	615	882-1244	Hahn	138967.35
503	615	123-5589	Okon	127093.45

Keys

- Candidate keys
  - CODE – minimal superkey
- Primary key
  - CODE – only possible candidate key to select
- Secondary key
  - LNAME + FNAME (possibly?)
- Surrogate key
  - None in this example
- Foreign key
  - Agent Code

Table name: CUSTOMER **Database name: Ch03_InsureCo**
Primary key: CUS_CODE
Foreign key: AGENT_CODE

CUS_CODE	CUS_LNAME	CUS_FNAME	CUS_INITIAL	CUS_RENEW_DATE	AGENT_CODE
10010	Ramas	Alfred	A	05-Apr-2018	502
10011	Dunne	Leona	K	16-Jun-2018	501
10012	Smith	Kathy	W	29-Jan-2019	502
10013	Olowski	Paul	F	14-Oct-2018	
10014	Orlando	Myron		28-Dec-2018	501
10015	O'Brian	Amy	B	22-Sep-2018	503
10016	Brown	James	G	25-Mar-2019	502
10017	Williams	George		17-Jul-2018	503
10018	Farriss	Anne	G	03-Dec-2018	501
10019	Smith	Olette	K	14-Mar-2019	503

Table name: AGENT (only five selected fields are shown)
Primary key: AGENT_CODE
Foreign key: none

AGENT_CODE	AGENT_AREACODE	AGENT_PHONE	AGENT_LNAME	AGENT_YTD_SLS
501	713	228-1249	Alby	132735.75
502	615	882-1244	Hahn	138967.35
503	615	123-5589	Okon	127093.45

Integrity Rules

- Entity Integrity
  - Requirement
    - All PK entries are unique, no PK attribute can be null
  - Purpose
    - Each row has a unique identity
    - FK values can properly reference PK key values
  - Example
    - No two invoices can have the same invoice number
    - The invoice number cannot be null
    - Any table referring to the invoice table can identify the invoice using the number

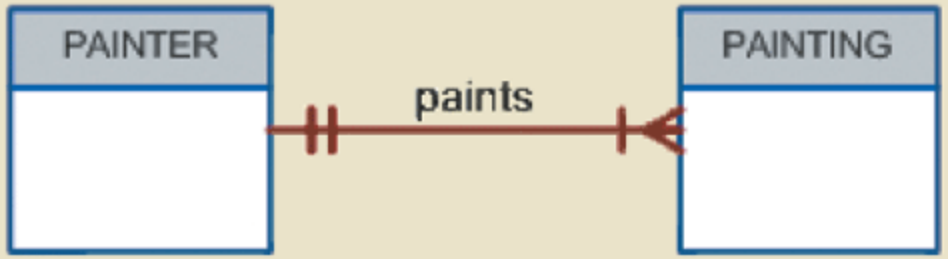
Integrity Rules

- Referential Integrity
  - Requirement
    - FK value matches an existing PK value of the related table or can be null
    - However, a FK cannot be null if it is a PK attribute of the table where it resides
  - Purpose
    - It is impossible to have an invalid FK value (i.e., it must reference an existing entity occurrence in the related table or be null)
    - Makes it impossible to delete the row being referenced by the FK
  - Example
    - A customer may not have a sales representative, but if it does, the sales representative must exist in the sales representative table
    - The sales representative cannot be removed from the sales representative table if a customer refers to it

Relationships – One-to-Many (1:M)

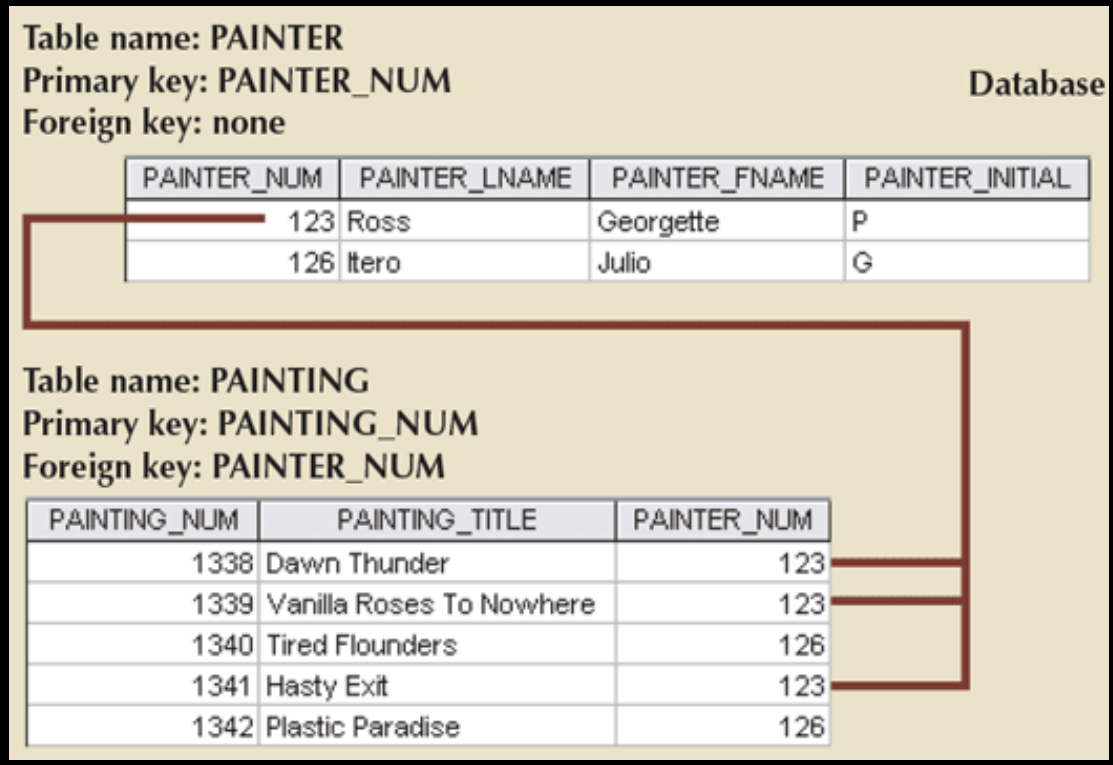
- One row of Table A links with many rows of Table B
- However, only one in row of Table B links to one row of Table A

FIGURE 3.17 THE 1:M RELATIONSHIP BETWEEN PAINTER AND PAINTING



Relationships – One-to-Many (1:M)

- PK of Painter
- PK of Painting
- FK of Painting
- In all cases of painter-paintings:
  - A painter paints many paintings
  - A painting is painted by one painter
- The PK from the “one-side” is placed as a FK in the “many-side.” Why?



Relationships – One-to-Many (1:M)

- The PK from the “one-side” is placed as a FK in the “many-side.” Why?

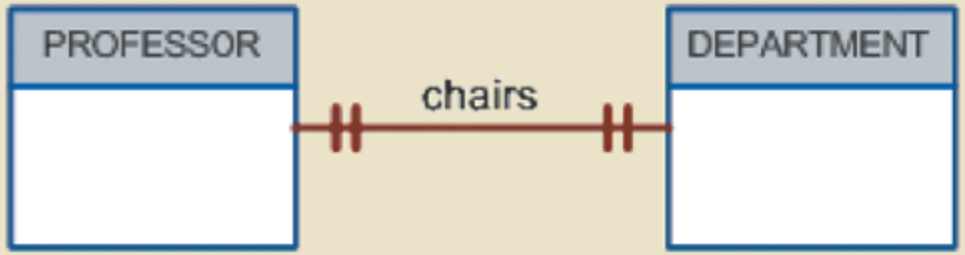
PAINTER_NUM	PAINTER_LNAME	PAINTER_LNAME	PAINTER_INITIAL	PAINTING1	PAINTING2	PAINTING3
123	Ross	Georgette	P	1338	1339	1341
126	Itero	Julio	G	1340	1342	

PAINTER_NUM	PAINTER_LNAME	PAINTER_LNAME	PAINTER_INITIAL	PAINTING
123	Ross	Georgette	P	1338
123	Ross	Georgette	P	1339
123	Ross	Georgette	P	1341
126	Itero	Julio	G	1340
126	Itero	Julio	G	1342

Relationships – One-to-One (1:1)

- One row of Table A links with only one row of Table B
- And only one in row of Table B links to only one row of Table A

FIGURE 3.21 THE 1:1 RELATIONSHIP BETWEEN PROFESSOR AND DEPARTMENT



Relationships – One-to-One (1:1)

- PK is dept_code
- FK is emp_num
- Why not have dept_code as the FK in the employee table instead?

Table name: DEPARTMENT
Primary key: DEPT_CODE
Foreign key: EMP_NUM

The 1:1 PROFESSOR chairs DEPARTMENT relationship is implemented through the placement of the EMP_NUM foreign key in the DEPARTMENT table.

DEPT_CODE	DEPT_NAME	SCHOOL_CODE	EMP_NUM	DEPT_ADDRESS	DEPT_EXTENSION
ACCT	Accounting	BUS	114	KLR 211, Box 52	3119
ART	Fine Arts	A&SCI	435	BBG 185, Box 128	2278
BIOL	Biology	A&SCI	387	AAK 230, Box 415	4117
CIS	Computer Info. Systems	BUS	209	KLR 333, Box 56	3245
ECON/FIN	Economics/Finance	BUS	299	KLR 284, Box 63	3126
ENG	English	A&SCI	160	DRE 102, Box 223	1004
HIST	History	A&SCI	103	DRE 156, Box 284	1867
MATH	Mathematics	A&SCI	297	AAK 194, Box 422	4234
MKT/MGT	Marketing/Management	BUS	106	KLR 126, Box 55	3342
PSYCH	Psychology	A&SCI	195	AAK 297, Box 438	4110
SOC	Sociology	A&SCI	342	BBG 208, Box 132	2008

EMP_NUM	PROF_OFFICE	PROF_EXTENSION	PROF_HIGH_DEGREE
105	DRE 102	8665	Ph.D.
114	KLR 211	4436	Ph.D.
301	KLR 244	4683	Ph.D.

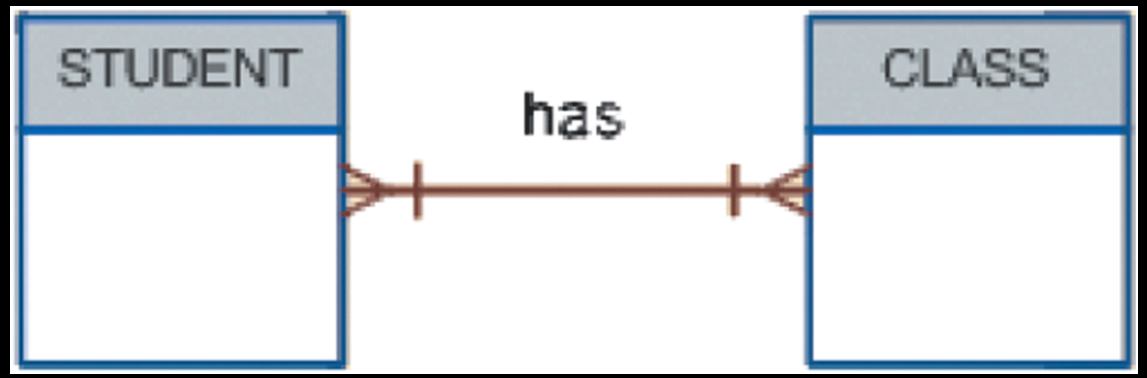
Relationships – One-to-One (1:1)

EMP_NUM	PROF_OFFICE	PROF_EXTENSION	PROF_HIGH_DEGREE	CHAIRS_DEPT
105	DRE 102	8665	Ph.D.	
114	KLR 211	4436	Ph.D.	ACCT
301	KLR 244	4683	Ph.D.	

- General rule: The PK from one table is placed in the other table to minimize redundant data and null entries in the FK

Relationships – Many-to-Many (M:N)

- Many rows of Table A links with many rows of Table B
- And many rows of Table B links to many rows of Table A



Relationships – Many-to-Many (M:N)

- PK of Student?!?
  - stu_num – not possible
  - Must be stu_num and class_code
- PK of Class?!?
  - class_code – not possible
  - Must be class_code and stu_num
  - But what if a class currently does not have any students?
- Results in a lot of redundant data, complicated composite keys, and complex queries
- Note the common attributes are stu_num and class_code

Table name: STUDENT
Primary key: STU_NUM
Foreign key: none
Database name: Ch03_C

STU_NUM	STU_LNAME	CLASS_CODE
321452	Bowser	10014
321452	Bowser	10018
321452	Bowser	10021
324257	Smithson	10014
324257	Smithson	10018
324257	Smithson	10021

Table name: CLASS
Primary key: CLASS_CODE
Foreign key: STU_NUM

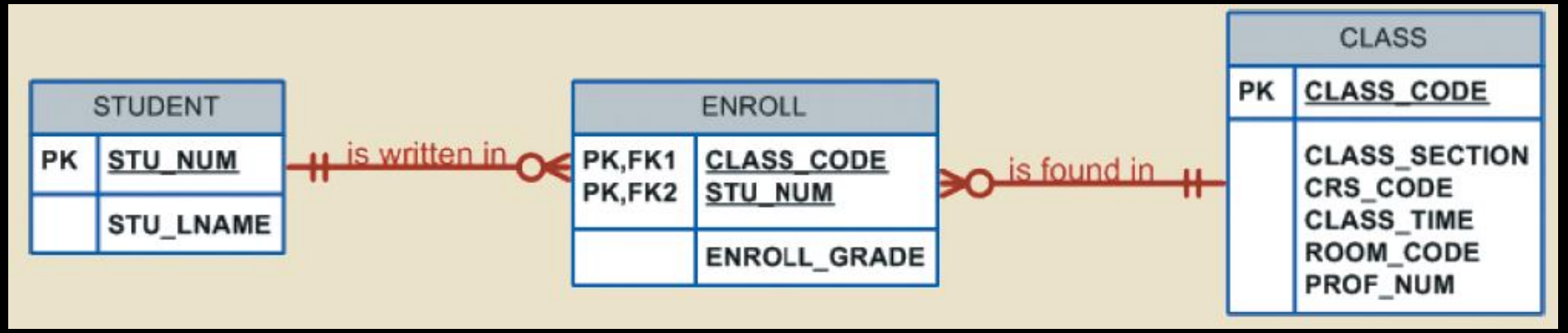
CLASS_CODE	STU_NUM	CRS_CODE	CLASS_SECTION	CLASS_TIME	CLASS_ROOM	PROF_NUM
10014	321452	ACCT-211	3	TTh 2:30-3:45 p.m.	BUS252	342
10014	324257	ACCT-211	3	TTh 2:30-3:45 p.m.	BUS252	342
10018	321452	CIS-220	2	MWF 9:00-9:50 a.m.	KLR211	114
10018	324257	CIS-220	2	MWF 9:00-9:50 a.m.	KLR211	114
10021	321452	QM-261	1	MWF 8:00-8:50 a.m.	KLR200	114
10021	324257	QM-261	1	MWF 8:00-8:50 a.m.	KLR200	114

Relationships – Many-to-Many (M:N)

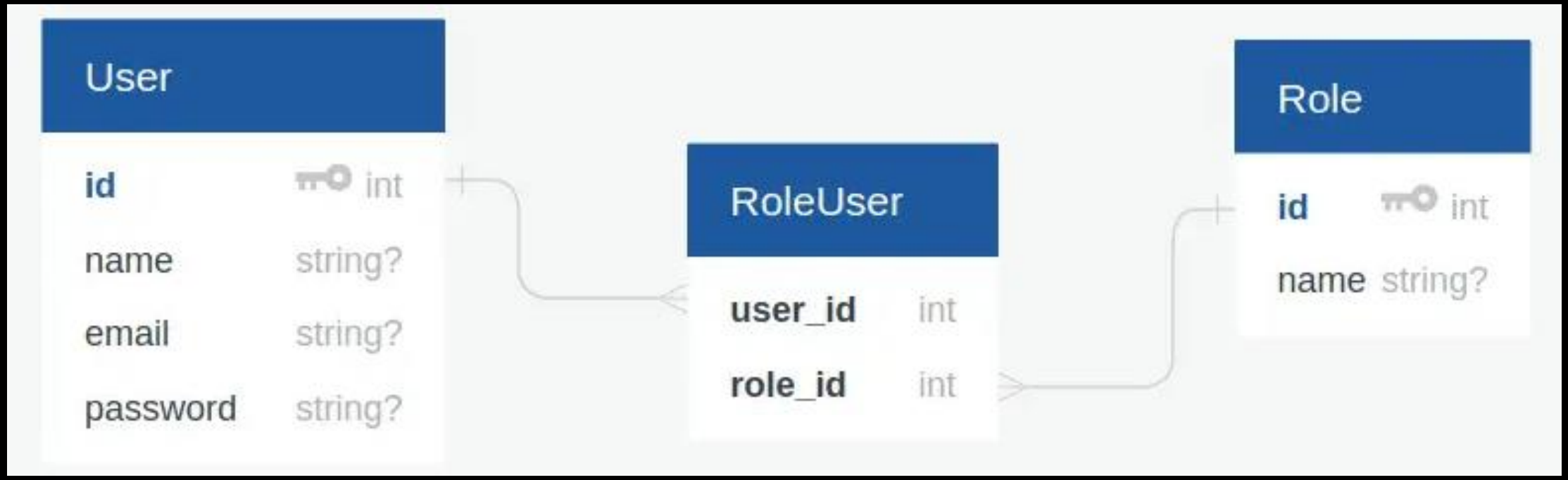
- Many-to-Many relationships are resolved with bridge entities
  - Bridge entities are also called associative or composite entities
  - Resolving the M:N
    - Form a bridge entity composed of the PK of each entity
    - The PK of the bridge entity is a composite primary key
    - Each PK attribute of the bridge entity is a FK to another table
    - Remember that the PK attribute cannot be null, so the FK attributes cannot be null
    - Suppose we allow a null FK, why does that not make sense intuitively?
  - You can have additional attributes in the bridge entities
    - These typical exist because of the relationship; they describe something about it
  - The M:N relationships still exists!

Relationships – Many-to-Many (M:N)

- Many students enroll in many class, and many classes are taken by many students—the M:N relationship is maintained
- *Important: Unless otherwise specified, all M:N relationships must be resolved in any work we complete for this course*



Relationships – Many-to-Many (M:N)



https://www.itsolutionstuff.com/post/laravel-many-to-many-eloquent-relationship-tutorialexample.html

Selecting Primary Keys

- Natural keys
  - Real world, generally accepted identifier
  - Familiar to end-users
- Primary key guidelines
  - The PK is to ensure entity integrity, not describe the entity
  - The PK is used to establish relationships unseen by the end user
- Desirable PK characteristics
  - Non-intelligent
  - No change over time
  - Preferably single-attribute
  - Preferably numeric
  - Security compliant

Selecting Primary Keys

- Composite keys
  - Used in bridge entities
  - Used in weak entities (we will see this later)
  - Not favored in other entities
- Surrogate keys
  - When there is no natural or suitable primary key
  - Good when the only other option are large composite keys
  - Good when the other at option does not meet the desirable characteristics