Dalhousie University
Faculty of Computer Science

CSCI 2141 – Intro to Database Systems

Week 3 – The Relational Database Model

Learning Objectives

- During this week, you will learn:
  - Keys and types of keys
  - Relational database operators
  - How data redundancy is handled in the relational database model

A Logical View of Data

- Relational database model enables logical representation of the data and its relationships
- Logical simplicity yields simple and effective database design methodologies
- Facilitated by the creation of data relationships based on a logical construct called a relation

Characteristics of a Relational Table

CHARACTERISTICS OF A RELATIONAL TABLE	
1	A table is perceived as a two-dimensional structure composed of rows and columns.
2	Each table row (**tuple**) represents a single entity occurrence within the entity set.
3	Each table column represents an attribute, and each column has a distinct name.
4	Each intersection of a row and column represents a single data value.
5	All values in a column must conform to the same data format.
6	Each column has a specific range of values known as the attribute **domain**.
7	The order of the rows and columns is immaterial to the DBMS.
8	Each table must have an attribute or combination of attributes that uniquely identifies each row.

Table name: STUDENT

STU_LNAME	STU_FNAME	STU_INIT	STU_DOB	STU_HRS	STU_CLASS	STU_GPA
Bowser	William	C	12-Feb-1985	42	So	2.84
Smithson	Anne	K	15-Nov-1991	81	Jr	3.27
Brewer	Juliette		23-Aug-1979	36	So	2.26
Oblonski	Walter	H	16-Sep-1986	66	Jr	3.09
Smith	John	D	30-Dec-1968	102	Sr	2.11
Katinga	Raphael	P	21-Oct-1989	114	Sr	3.15
Robertson	Gerald	T	08-Apr-1983	120	Sr	3.87
Smith	John	B	30-Nov-1996	15	Fr	2.92

Identifying a Row

- In the table below, what pieces of data would you need to find the GPA of a specific student?

Table name: STUDENT

STU_LNAME	STU_FNAME	STU_INIT	STU_DOB	STU_HRS	STU_CLASS	STU_GPA
Bowser	William	C	12-Feb-1985	42	So	2.84
Smithson	Anne	K	15-Nov-1991	81	Jr	3.27
Brewer	Juliette		23-Aug-1979	36	So	2.26
Oblonski	Walter	H	16-Sep-1986	66	Jr	3.09
Smith	John	D	30-Dec-1968	102	Sr	2.11
Katinga	Raphael	P	21-Oct-1989	114	Sr	3.15
Robertson	Gerald	T	08-Apr-1983	120	Sr	3.87
Smith	John	B	30-Nov-1996	15	Fr	2.92

- STU_LNAME, STU_FNAME, STU_INIT, STU_DOB

Identifying a Row

- How about this table? What pieces of data would you need to find the GPA of a specific student?

Table name: STUDENT

STU_NUM	STU_LNAME	STU_FNAME	STU_INIT	STU_DOB	STU_HRS	STU_CLASS	STU_GPA
321452	Bowser	William	C	12-Feb-1985	42	So	2.84
324257	Smithson	Anne	K	15-Nov-1991	81	Jr	3.27
324258	Brewer	Juliette		23-Aug-1979	36	So	2.26
324269	Oblonski	Walter	H	16-Sep-1986	66	Jr	3.09
324273	Smith	John	D	30-Dec-1968	102	Sr	2.11
324274	Katinga	Raphael	P	21-Oct-1989	114	Sr	3.15
324291	Robertson	Gerald	T	08-Apr-1983	120	Sr	3.87
324299	Smith	John	B	30-Nov-1996	15	Fr	2.92

- STU_NUM

Keys

- Consist of one or more attributes that determine other attributes
- Used to:
  - Ensure that each row in a table is uniquely identifiable
  - Establish relationships among tables and to ensure the integrity of the data

What is Determination?

- The role of key is based on the concept of determination
- Determination is the state in which knowing the value of one attribute makes it possible to determine the value of another
  - In database environments, determination is based on the relationships among attributes
  - Consider the following example:

Stu_Num	Stu_Lname	Stu_GPA
321452	Bowser	2.84
324257	Smithson	3.27

    - Knowing which of the above attributes, can you determine the others?
    - Knowing Stu_Num, you can determine the Stu_Lname and Stu_GPA

Functional Dependence

- A specific terminology and notation is used to describe relationships based on determination
  - The relationship is called **Functional Dependence**
    - The value of one (or more) attributes, determine the value of one or more other attributes
    - In the previous example, Student_ID attribute determines the value of Student_Lname and Student_GPA attributes
- The attribute whose value determines another is called the **determinant** or the key
- The attribute whose value is determined by the other is called a **dependent**

Functional Dependence

Student_ID	Student_Lname	Student_GPA

- In the above example, identify the determinant and the dependent
  - Determinant: | Student_ID |
  - Dependents: | Student_Lname | Student_GPA |
- Functional dependence is shown as follows:
  - **Student_ID → (Student_Lname, Student_GPA)**

Functional Dependence

- Determinants may contain more than one attribute
  - It may be possible to have a functional dependency in which the determinant contains attributes that are not necessary for the relationship
  - Consider the following functional dependencies

    Student_ID → Student_GPA

    (Student_ID, Student_Lname) → (Student_GPA)

    - Clearly, the Student_Lname attribute is not necessary for the relationship
      - The functional dependency is valid because a pair of values for Student_ID and Student_Lname will return only one value for Student_GPA

Full Functional Dependence

- Functional dependency in which the entire collection of attributes in the determinant is necessary for the relationship is called a **full functional dependence**
  - Functional dependency

    (Student_ID, Student_Lname) → (Student_GPA)
  - Full functional dependency

    Student_ID → Student_GPA

Dependencies

- In the tables shown earlier, can you identify some dependencies?
- In the first table:

(STU_LNAME, STU_FNAME, STU_INIT, STU_DOB) → (STU_HRS, STU_GPA)

Determinant → Dependent

- In the second table:

STU_NUM → (STU_LNAME, STU_FNAME, STU_GPA)

- Can you identify the type of dependencies?
- How about this:

(STU_NUM, STU_LNAME) → (STU_GPA)

Functional Dependence – Examples

- Cus_code → (Cus_Lname, Cus_Fname, Cus_Phone, Cus_dob)
- (Cus_code, Cus_Lname) → (Cus_Fname, Cus_Phone, Cus_dob)
- Cus_Phone → (Cus_code, Cus_Lname, Cus_Fname, Cus_dob)

How about?

- (Cus_Lname, Cus_Fname, Cus_dob) → (Cus_code, Cus_Phone)
  - Probably not, as two people with same name may have the same dob!

CUS_CODE	CUS_LNAME	CUS_FNAME	CUS_PHONE	CUS_DOB
10010	Ramas	Alfred	844-2573	2000-01-01
10011	Dunne	Leona	894-1238	2000-01-01
10012	Smith	Kathy	894-2285	2000-01-01
10013	Olowski	Paul	894-2180	2000-01-01

Keys

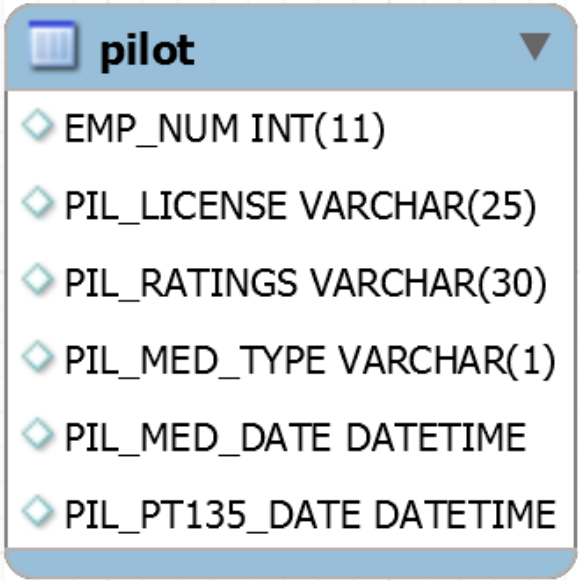
- **Super key**: Key that can uniquely identify any row in the table
  - Determinant of a functional dependency
- **Candidate key**: is a minimal super key
  - Determinant of a full functional dependency
- **Primary key (PK)**: Attribute or combination of attributes that uniquely identifies any given row
  - Chosen from among the candidate keys

Keys

- What key would the following attributes form?
  - (Student_ID, Student_Lname, Student_GPA) ***Super key***
  - (Student_ID, Student_Lname) ***Super key***
  - (Student_Lname, Student_GPA) ***None***
  - (Student_ID, Student_GPA) ***Super key***
  - Student_Lname ***None***
  - Student_ID ***Super, candidate and primary key***

Keys – Example

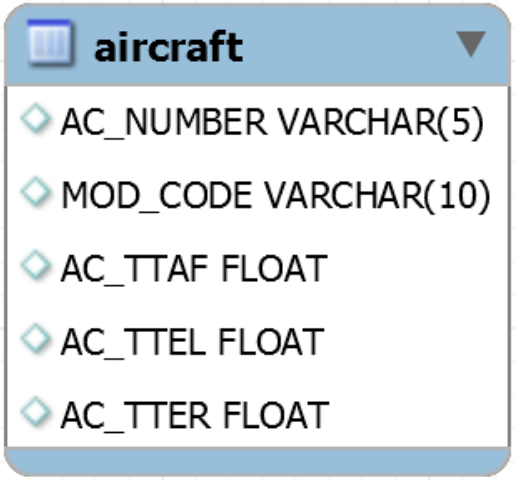
- Identify Superkey(s)
  - (EMP_NUM, PIL_LICENSE)
  - Any combination with EMP_NUM
- Candidate key(s)
  - EMP_NUM
  - No other combination that could be suitable
- Primary key
  - EMP_NUM



EMP_NUM	PIL_LICENSE	PIL_RATINGS	PIL_MED_TYPE	PIL_MED_DATE	PIL_PT135_DATE
101	ATP	ATP/SEL/MEL/Instr/CFII	1	2016-01-20 00:00:00	2016-01-11 00:00:00
104	ATP	ATP/SEL/MEL/Instr	1	2015-12-18 00:00:00	2016-01-17 00:00:00
105	COM	COMM/SEL/MEL/Instr/CFI	2	2016-01-05 00:00:00	2016-01-02 00:00:00
106	COM	COMM/SEL/MEL/Instr	2	2015-12-10 00:00:00	2016-02-02 00:00:00
109	COM	ATP/SEL/MEL/SES/Instr/CFII	1	2016-01-22 00:00:00	2016-01-15 00:00:00

Keys – Example

- Identify Superkey(s)
  - (AC_NUMBER, MOD_CODE)
  - Any combination with AC_NUMBER
- Candidate key(s)
  - AC_NUMBER
  - No other combination that could be suitable
- Primary key
  - AC_NUMBER

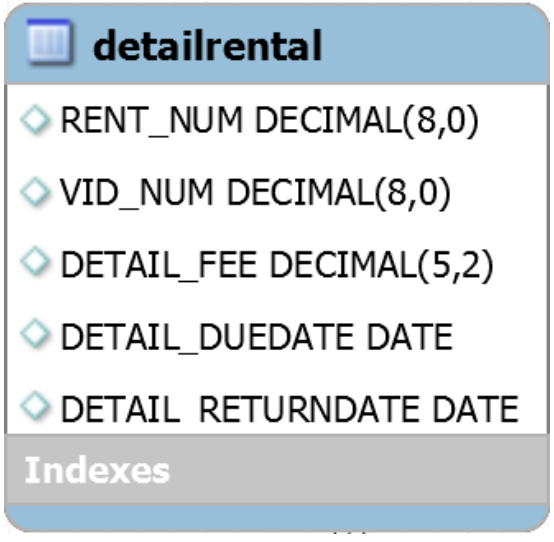


AC_NUMBER	MOD_CODE	AC_TTAF	AC_TTEL	AC_TTER
1484P	PA23-250	1833.1	1833.1	101.8
2289L	C-90A	4243.8	768.9	1123.4
2778V	PA31-350	7992.9	1513.1	789.5
4278Y	PA31-350	2147.3	622.1	243.2

TTAF – Total Time Air Frame
TTEL – Total Time Engine Left
TTER – Total Time Engine Right

Keys – Example

- Identify Superkey(s)
  - (RENT_NUM, VID_NUM, DETAIL_FEE)
  - Any combination with RENT_NUM, VID_NUM
- Candidate key(s)
  - (RENT_NUM, VID_NUM)
  - No other combination that could be suitable
- Primary key
  - (RENT_NUM, VID_NUM)



RENT_NUM	VID_NUM	DETAIL_FEE	DETAIL_DUEDATE	DETAIL_RETURNDATE
1001	34342	2.00	2016-03-04	2016-03-02
1001	34366	3.50	2016-03-04	2016-03-02
1001	61353	2.00	2016-03-04	2016-03-03
1002	59237	3.50	2016-03-04	2016-03-04
1003	54325	3.50	2016-03-04	2016-03-09

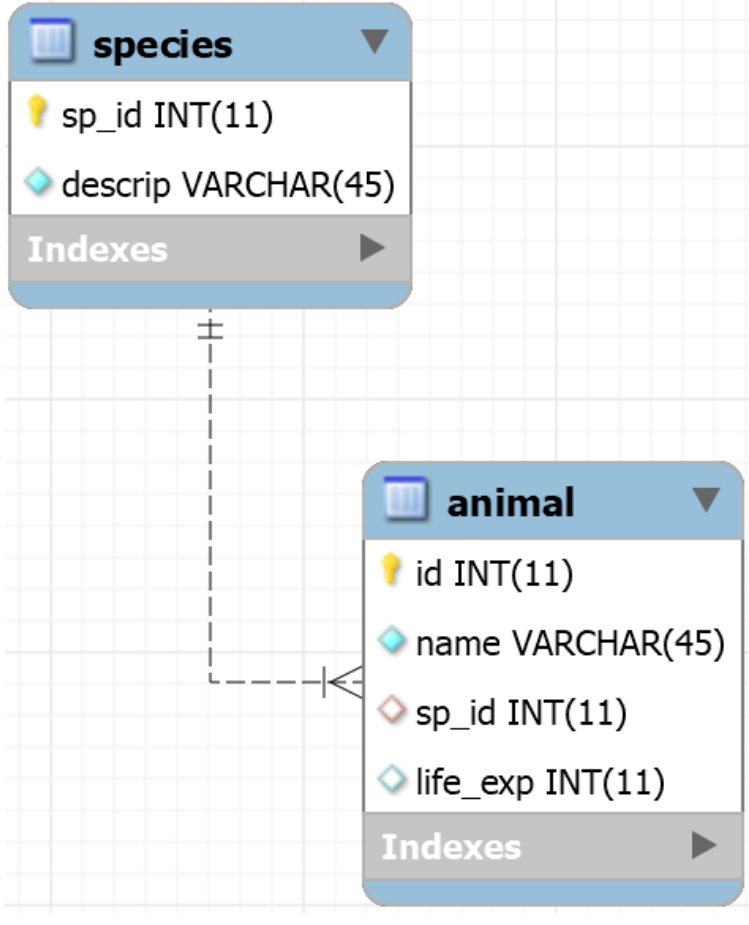
Some Key Terminology

- **Key attribute**: Attribute that is a part of a key
- **Entity integrity**: Condition in which each row in the table has its own unique identity
  - All of the values in the primary key must be unique
  - No key attribute in the primary key can contain a null
- **Null**: Absence of any data value that could represent:
  - An unknown attribute value
  - A known, but missing, attribute value
  - A inapplicable condition
- **Referential integrity**: Every reference to an entity instance by another entity instance is valid

An Example of a Simple Relational Database

sp_id	descrip
1	INSECT
2	BIRD
3	FISH
4	MAMMAL

id	name	sp_id	life_exp
1	Cat	4	20
2	Elephant	4	70
3	Trout	3	5
4	Shark	3	25
5	Canary	2	20
6	Albatross	2	40
7	Swift	2	5



An Example of a Simple Relational Database

Table name: PRODUCT
Primary key: PROD_CODE
Foreign key: VEND_CODE

PROD_CODE	PROD_DESCRIPT	PROD_PRICE	PROD_ON_HAND	VEND_CODE
001278-AB	Claw hammer	12.95	23	232
123-21UUY	Houselite chain saw, 16-in. bar	189.99	4	235
QER-34256	Sledge hammer, 16-lb. head	18.63	6	231
SRE-657UG	Rat-tail file	2.99	15	232
ZZX/3245Q	Steel tape, 12-ft. length	6.79	8	235

link

Table name: VENDOR
Primary key: VEND_CODE
Foreign key: none

VEND_CODE	VEND_CONTACT	VEND_AREACODE	VEND_PHONE
230	Shelly K. Smithson	608	555-1234
231	James Johnson	615	123-4536
232	Annelise Crystall	608	224-2134
233	Candice Wallace	904	342-6567
234	Arthur Jones	615	123-3324
235	Henry Ortozo	615	899-3425

Integrity Rules

- Entity Integrity
  - All PK entries are unique, and no part of a PK may be null
  - Each row will have a unique identity, and foreign key (FK) values can properly reference PK values
  - Example: No student can have a duplicate Registration number
- Referential Integrity
  - FK may or may not be part of its table's PK
    - FK may be null, if it is not part of the PK
  - Every non-null FK value must reference an existing PK value in the table to which it is related
    - Example: A student may not be registered in any courses, but it will be impossible to register the student in a course with invalid course code

Integrity Rules – Example

Table name: CUSTOMER
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-2016	502
10011	Dunne	Leona	K	16-Jun-2016	501
10012	Smith	Kathy	W	29-Jan-2017	502
10013	Olowski	Paul	F	14-Oct-2016	
10014	Orlando	Myron		28-Dec-2016	501
10015	O'Brian	Amy	B	22-Sep-2016	503
10016	Brown	James	G	25-Mar-2017	502
10017	Williams	George		17-Jul-2016	503
10018	Farriss	Anne	G	03-Dec-2016	501
10019	Smith	Olette	K	14-Mar-2017	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

Column Constraints

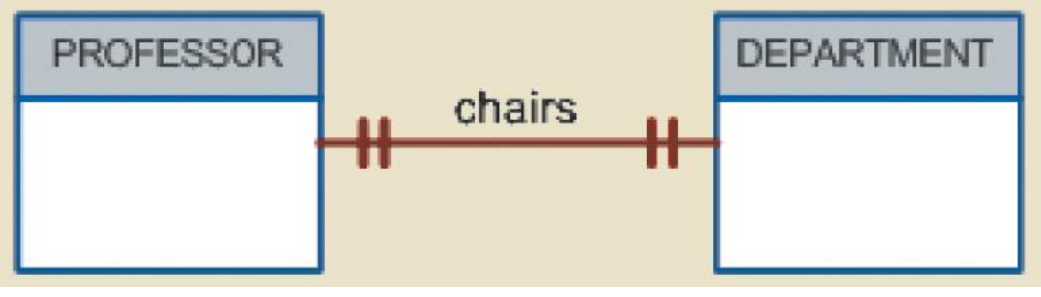
- Some constraints that can be put on a column are:
  - NOT NULL constraint - Placed on a column to ensure that every row in the table has a value for that column
  - UNIQUE constraint - Restriction placed on a column to ensure that no duplicate values exist for that column

Relationships

Relationships within the Relational Database

- 1:M relationship - Norm for relational databases
- 1:1 relationship - One entity can be related to only one other entity and vice versa
- Many-to-many (M:N) relationship - Implemented by creating a new entity in 1:M relationships with the original entities
  - **Composite entity** (**Bridge** or **associative entity**): Helps avoid problems inherent to M:N relationships, includes the primary keys of tables to be linked

The 1:1 Relationship



M:N Relationship

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

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

Changing the M:N Relationship to Two 1:M Relationships

Table name: STUDENT

STU_NUM	STU_LNAME
321452	Bowser
324257	Smithson

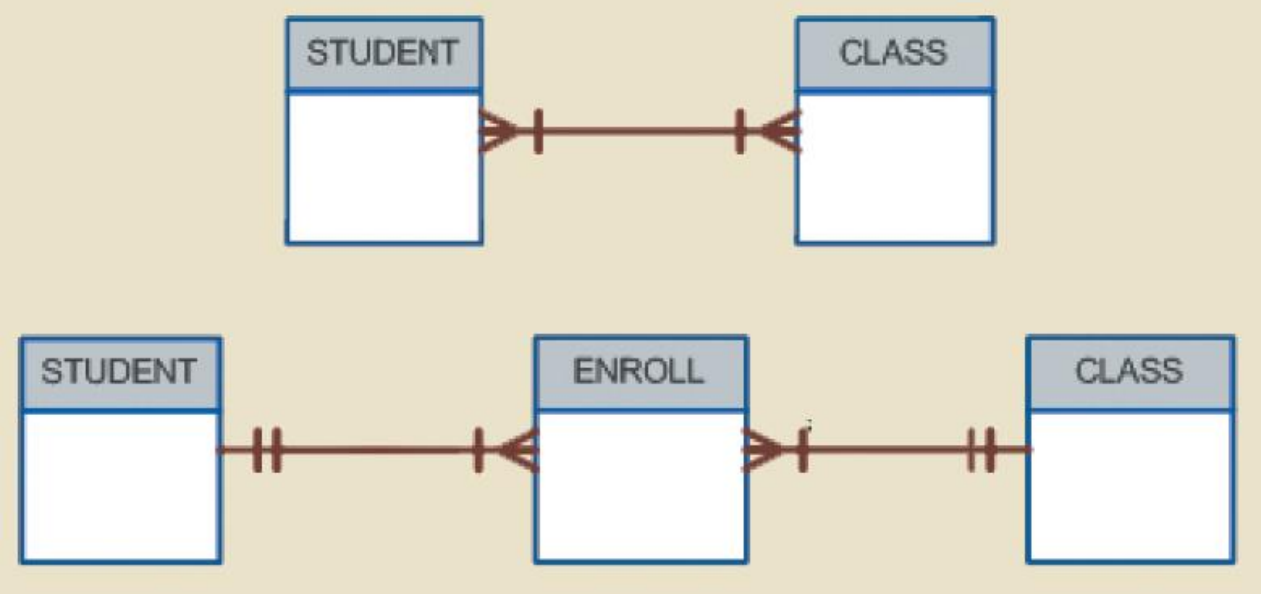
Table name: ENROLL

CLASS_CODE	STU_NUM	ENROLL_GRADE
10014	321452	C
10014	324257	B
10018	321452	A
10018	324257	B
10021	321452	C
10021	324257	C

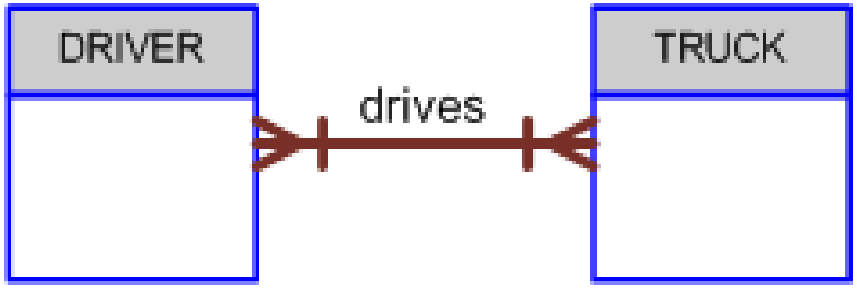
Table name: CLASS

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

Changing the M:N Relationship to Two 1:M Relationships



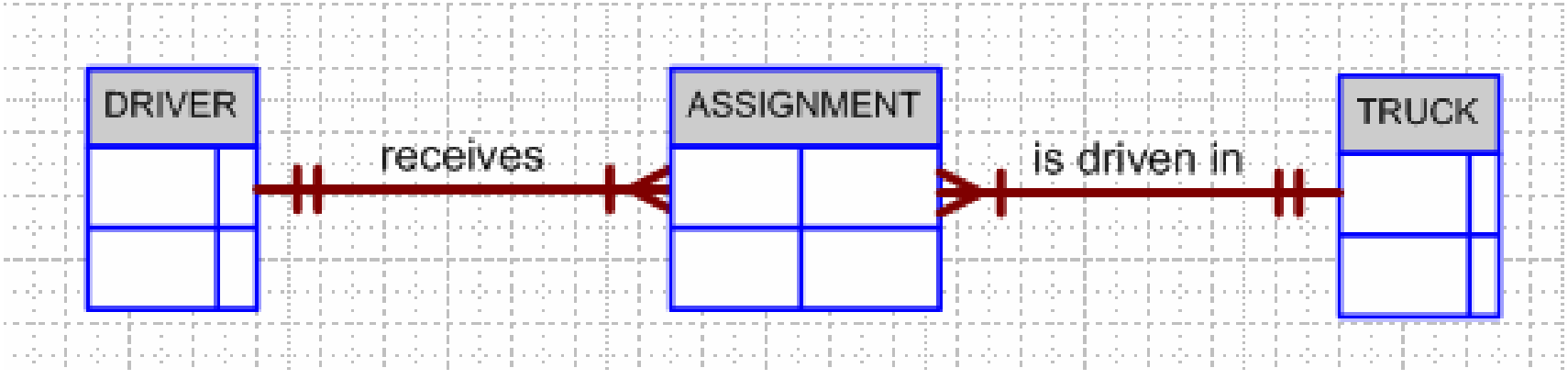
Exercise 2 – Many-to-many Relationship



During some time interval, a DRIVER can drive many TRUCKs and any TRUCK can be driven by many DRIVERs

Convert the above diagram into an Entity-Relationship Model (ERM) using only 1-to-M relationships

Solution – Exercise 2



- 1 to M relationship is based on the following business rules
  - A driver may receive many (driving) assignments.
  - Each (driving) assignment is made for a single driver.
  - A truck may be driven in many (driving) assignments.
  - Each (driving) assignment is made for a single truck.

Redundancy

- If you delete an attribute and the original information can still be generated through relational algebra, that attribute is redundant

- Test of redundancy
  - whether elimination of an attribute will eliminate information

Data Redundancy Revisited

- Relational database facilitates control of data redundancies through use of foreign keys
- To be controlled except the following circumstances
  - Data redundancy must be increased to make the database serve crucial information purposes
  - Exists to preserve the historical accuracy of the data

Data Redundancy Revisited

Table name: CUSTOMER

CUS_CODE	CUS_LNAME	CUS_FNAME	CUS_INITIAL	CUS_AREACODE	CUS_PHONE
10010	Ramas	Alfred	A	615	844-2573
10011	Dunne	Leona	K	713	894-1238
10012	Smith	Kathy	W	615	894-2285
10013	Olowski	Paul	F	615	894-2180
10014	Orlando	Myron		615	222-1672
10015	O'Brian	Amy	B	713	442-3381
10016	Brown	James	G	615	297-1228
10017	Williams	George		615	290-2556
10018	Farriss	Anne	G	713	382-7185
10019	Smith	Olette	K	615	297-3809

Table name: INVOICE

INV_NUMBER	CUS_CODE	INV_DATE
1001	10014	08-Mar-16
1002	10011	08-Mar-16
1003	10012	08-Mar-16
1004	10011	09-Mar-16

Table name: PRODUCT

PROD_CODE	PROD_DESCRIPT	PROD_PRICE	PROD_ON_HAND	VEND_CODE
001278-AB	Claw hammer	12.95	23	232
123-21UUY	Houselite chain saw, 16-in. bar	189.99	4	235
QER-34256	Sledge hammer, 16-lb. head	18.63	6	231
SRE-657UG	Rat-tail file	2.99	15	232
ZZX/3245Q	Steel tape, 12-ft. length	6.79	8	235

Table name: LINE

INV_NUMBER	LINE_NUMBER	PROD_CODE	LINE_UNITS	LINE_PRICE
1001	1	123-21UUY	1	189.99
1001	2	SRE-657UG	3	2.99
1002	1	QER-34256	2	18.63
1003	1	ZZX/3245Q	1	6.79
1003	2	SRE-657UG	1	2.99
1003	3	001278-AB	1	12.95
1004	1	001278-AB	1	12.95
1004	2	SRE-657UG	2	2.99

The Relational Diagram for the Invoicing System