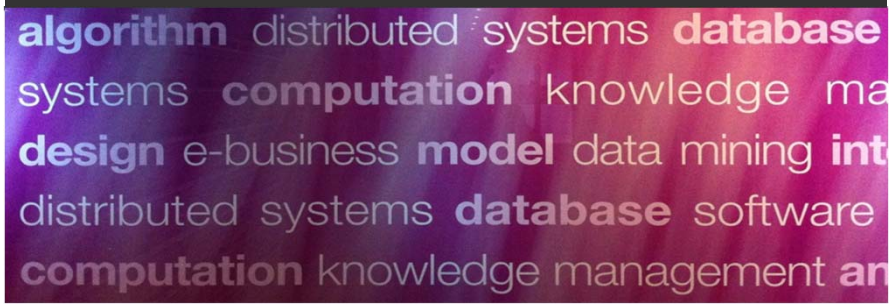
MONASH University

Information Technology

FIT3176 Advanced Database Design

Dr. Minh Le
Minh.Le@monash.edu

Topic 2: Advanced Data Modelling



*Adapted from slides developed by Lindsay Smith

Seek assistance as a preventative measure

Take the following relevant preventative measures as soon as possible, if you are falling behind in your studies:

- Study difficulties: Discuss any difficulties you are experiencing with your course leader, unit coordinator, lecturer or tutor.
  - These staff members can assist you in identifying your problem areas and explore the options available to you in your course.
- Language and learning online can help you with study methods, language skills and work presentation
  http://www.monash.edu.au/lls/llonline/
- Student life and support services can be found at:
  http://monash.edu/students/support/
- Study skills for University:
  http://www.monash.edu/rlo/study-skills

Do you have a medical condition or incapacity that impacts on your ability to effectively study?

Registered students are provided with a range of services including:

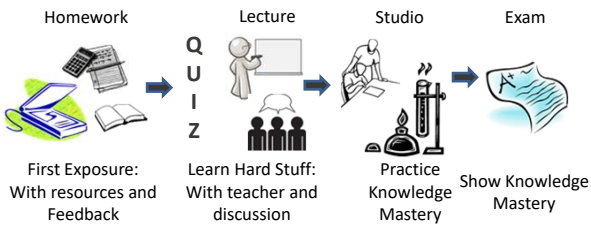
- Notetakers and Auslan interpreters
- Readings in alternative formats
- Adaptive equipment and software
- Alternative arrangements for exams

I had a broken wrist and I registered to have a Notetaker to take notes in my lectures and tutorials. This service was excellent

For further information and details about how to register:

Email: disabilityservices@monash.edu
Phone: 03 9905 5704
Web: monash.edu/social-justice/disability

Peer Instruction



- Using MARS
  1. Visit http://mars.mu on your internet enabled device
  2. Log in using your Authcate details (not required if you're already logged in to Monash)
  3. Touch the + symbol
  4. Enter the code for FIT3176: **YE10IP**
  5. Answer questions when they pop up.

Learning Objectives and References

Learning Objectives

By the end of this week you should be able to:

- explain the advantages gained by the use of supertype and subtypes
- model using an Extended Entity Relationship Diagram (EERD)
  - entity supertype and subtypes
  - entity clusters
- select an appropriate primary key for an entity, understanding the role of
  - natural keys
  - composite keys
  - surrogate keys
- formulate appropriate models for dealing with
  - 1:1 relationships
  - maintaining a history of time-variant data
  - fan traps, and
  - redundant relationships

Learning Objectives and References

References:

- Coronel & Morris, Database Systems: Design, Implementation & Management, 12th Edition 2015, Thomson Course Technology. Chapter 5
- Coronel, Morris, & Rob, Database Systems: Design, Implementation & Management, 11th Edition 2011, Thomson Course Technology. Chapter 5

NULL – Revision

- NULL is NOT a value - is a representation of the fact that there is NO VALUE
- Reasons for a NULL:
  - VALUE NOT APPLICABLE -
    - EMP relation - empno, deptno, salary, commission
      - commission only applies to staff in sales dept
  - VALUE UNKNOWN -
    - Joe's salary is NULL, Joe's salary is currently unknown
  - VALUE DOES NOT EXIST -
    - Tax File Number - is applicable to all employees BUT Joe may not have a number at this time
  - VALUE UNDEFINED -
    - Certain items explicitly undefined eg. divide by zero
      - Columns Number_of_payments, Total_payments
      - Column Average_payment_made
      - If Number_of_payments = 0 => Average undefined

NULL – Revision

- Most relational systems implement NULLs as VALUE UNKNOWN (UNK), which can lead to problems:
  - Joe's wage = salary + commission
    - commission UNK => wage UNK
  - 4 Customers - balance $100 $200 $300 NULL
    - NULL ignored for inbuilt SUM and AVG functions (treated as UNK)
    - tuple counted by COUNT function (mathematically AVG = SUM/ROWS)
    - AVG $200 if SUM $600, however, number of tuples = 4?
- Now have values -
  - TRUE, FALSE, UNKNOWN
  - called THREE-VALUED LOGIC

FlyRight Aircraft Maintenance (FRAM)

The FlyRight Aircraft Maintenance (FRAM) division of the FlyRight Company (FRC) performs all maintenance for FRC's aircraft.

FRAM have identified the following business rules:

- All mechanics are FRC employees. Not all employees are mechanics.
- Mechanics are specialists in a particular area. Some mechanics are specialised in engine (EN) maintenance. Some mechanics are specialised in airframe (AF) maintenance. Some mechanics are specialised in avionics (AV) maintenance. (Avionics are the electronic components of an aircraft that are used in communication and navigation.)
- All mechanics take periodic refresher courses to stay current in their areas of expertise. FRC tracks all courses taken by each mechanic - date, course type, certification (Y/N), and performance.

Q1. Storing the FlyRight Aircraft Maintenance (FRAM) employee details in a *single* table called EMPLOYEE

a. Would be an optimal solution with no design/storage issues
b. Would result in unnecessary nulls present in some rows
c. Could result in some rows not being able to be stored due to missing data
d. Could result in potentially invalid participation in relationship/s by some employee instances
e. B and C
f. B and D

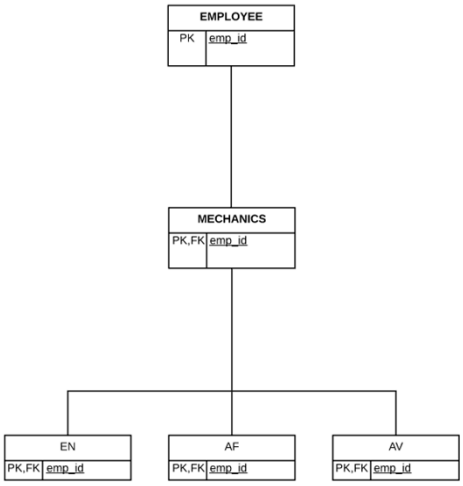
Entity Supertypes and Subtypes

- Entity supertype
  - Generic entity type related to one or more entity subtypes
  - Contains common characteristics
- Entity subtype
  - Contains unique characteristics of each entity subtype
- Provides two important benefits:
  - Avoids unnecessary nulls
  - Allows subtype to participate in relationships unique to that subtype (e.g., enabling a particular employee type to participate in relationships that are unique to that employee type.). That is, for example, Professor is a subtype of Employee supertype, and Professor can teach Class, while HRStaff is a subtype of Employee and HRStaff can manage Hiring.

Q2. Representing the FlyRight Aircraft Maintenance (FRAM) employees on a conceptual model in an optimal manner

a. Does not require the use of a supertype/subtype hierarchy
b. Requires one level of specialisation hierarchy:
   employees – mechanics
c. Requires two levels of specialisation hierarchy:
   employees – mechanics – mechanic types
d. Requires three levels of specialisation hierarchy:
   employees – mechanics – mechanic types – training
e. Would make use of a 1:M supertype/subtype hierarchy
f. More than one of the above answers are correct

FRAM 'starting' hierarchy



Specialization Hierarchy

- Depicts arrangement of higher-level entity supertypes and lower-level entity subtypes
- Relationships described in terms of "IS-A" relationships
- Subtype exists only within context of supertype
- **Every subtype has only one supertype** to which it is directly related
  - Subtype inherit their
    - primary key from their supertype
    - attributes and relationships of the supertype
  - Relationship is 1:1
- **Can have many levels of supertype/subtype relationships**
- Depicted in an Extended Entity Relationship Model using an Extended Entity Relationship Diagram (EERD)

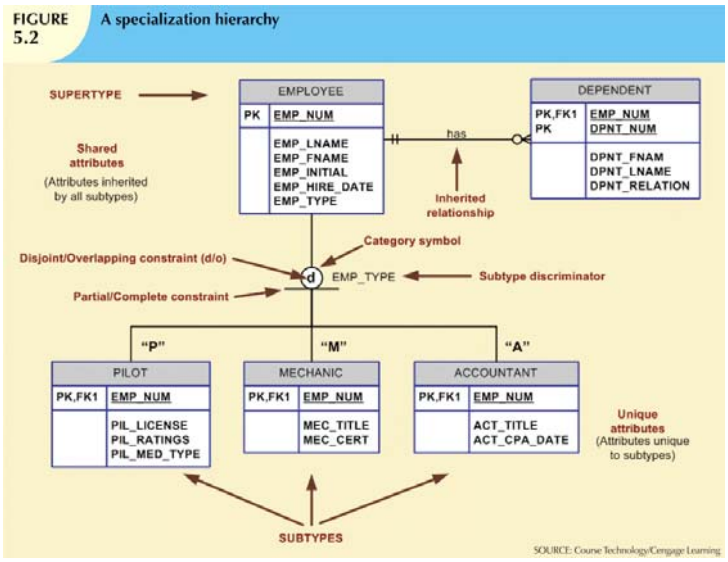
Subtype Discriminator & Constraints

- **Discriminator**
  - **Attribute in supertype** entity
    - Determines to which entity subtype each supertype occurrence is related
    - Default comparison condition for subtype discriminator attribute is equality comparison
- **Disjoint/Overlapping Constraints**
  - Disjoint subtypes
    - Also called non overlapping subtypes
    - Subtypes that contain unique subset of supertype entity set
  - Overlapping subtypes
    - Subtypes that contain non unique subsets of supertype entity set
    - Need multiple subtype discriminators

Completeness Constraint

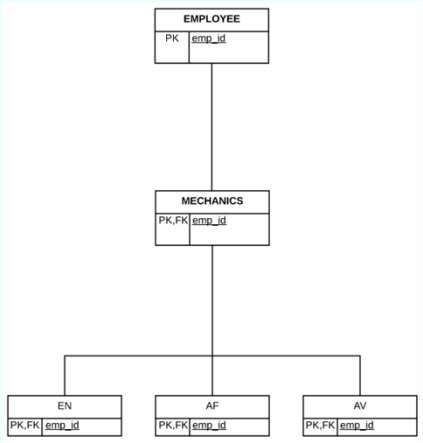
- Specifies whether entity supertype occurrence must be a member of at least one subtype
- Partial completeness
  - Symbolized by a circle over a single line
  - Some supertype occurrences are not members of any subtype
- Total completeness
  - Symbolized by a circle over a double line
  - Every supertype occurrence must be member of at least one subtype

Required layout for exam



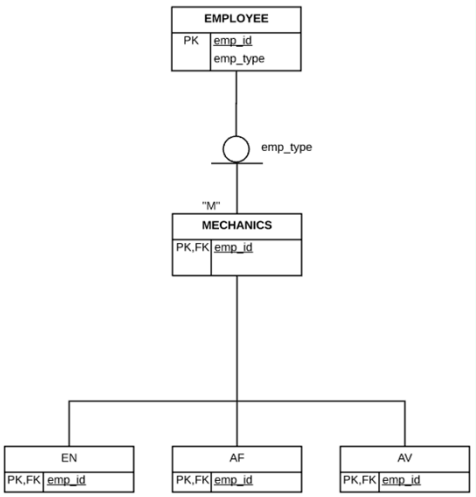
Q3. Employee to mechanic is represented by

a. Partial completeness
b. Total completeness
c. Disjoint subtypes
d. Overlapping subtypes
e. A and C
f. A and D
g. B and C
h. B and D

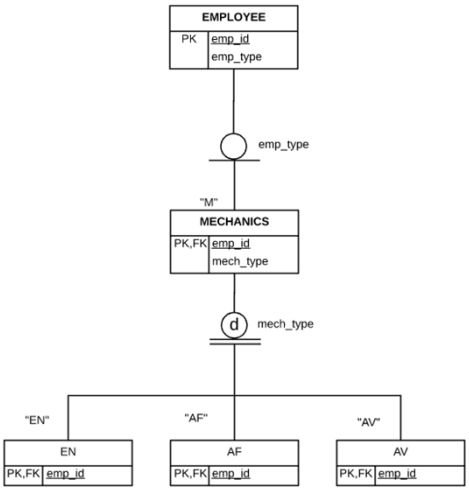


Q4. Mechanics to Mechanic type is represented by

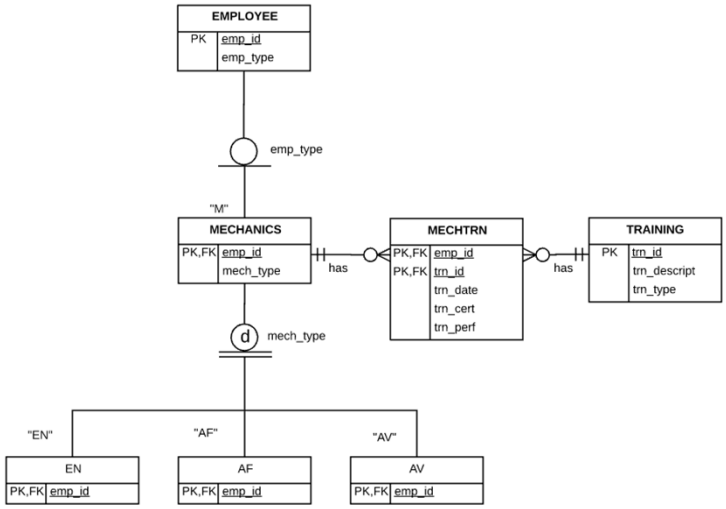
a. Partial completeness
b. Total completeness
c. Disjoint subtypes
d. Overlapping subtypes
e. A and C
f. A and D
g. B and C
h. B and D



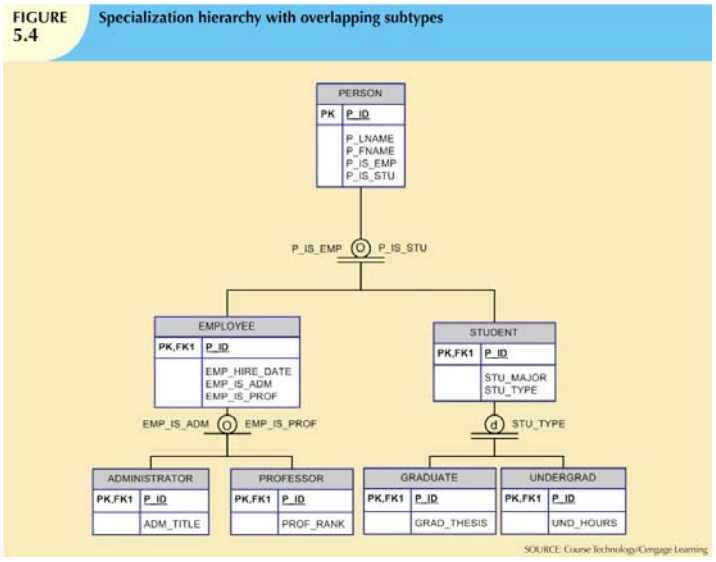
FRAM specialisation hierarchy



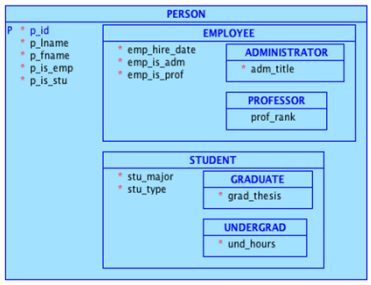
FRAM complete model

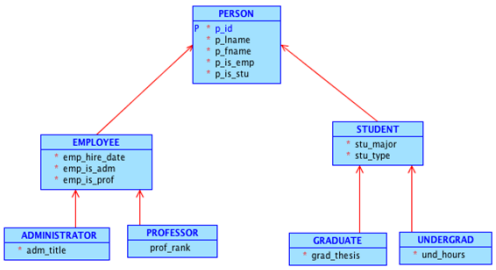


Required layout for exam



SQL Developer Logical Model





Box–in-Box Presentation

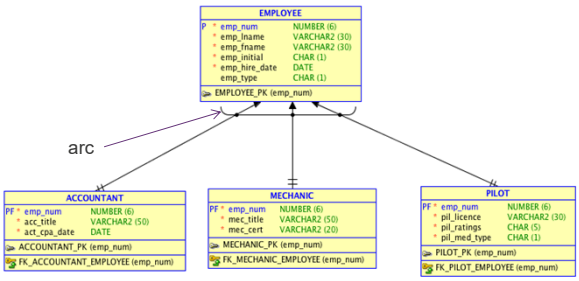
Required layout for FIT3176

Non Box-in-Box

Note PK is not added *to the sub-types*, present due to sub-super type relationship

SQL Developer Data Modeler – Relational Model

- SDDM currently only directly supports exclusive (disjoint) subtypes
  - Supported via an ARC .."An arc is an exclusive relationship group, which is defined such that only one of the relationships can exist for any instance of an entity."



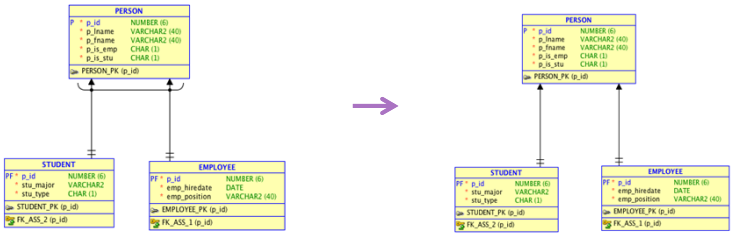
SQL Developer Data Modeler cont'd

- The arc will cause PL/SQL code to be generated of the form (*you will understand this code in a few weeks*):

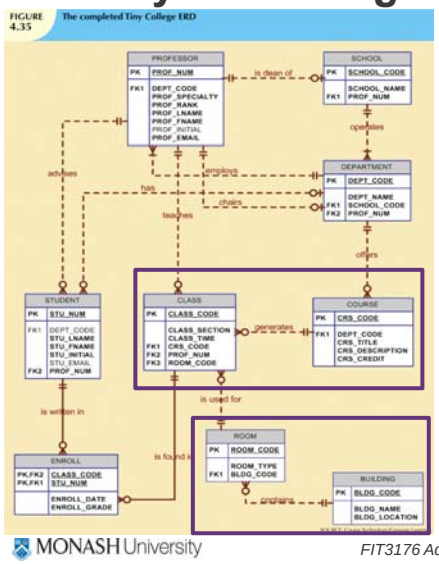
```
CREATE OR REPLACE TRIGGER ARC_FKArc_1_PILOT BEFORE
  INSERT
  OR
  UPDATE
    OF emp_num ON PILOT FOR EACH ROW DECLARE d VARCHAR2(1);
  BEGIN
    SELECT
      A.emp_type
    INTO
      d
    FROM
      EMPLOYEE A,
      PILOT B
    WHERE
      A.emp_num = B.emp_num ;
    IF (d       IS NULL OR d <> 'P') THEN
      raise_application_error(-20223,
      'FK PILOT.FK_PILOT_EMPLOYEE in Table PILOT violates Arc constraint
       on Table EMPLOYEE - only one FK is permitted'
      );
    END IF;
  END;
```

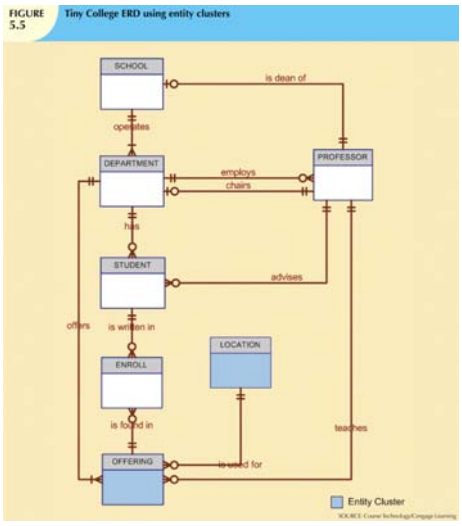

SQL Developer Data Modeler cont'd

- For overlap scenarios the arc **must be** deleted in the relational model, since SDDM does not support overlap



Entity Clustering





Concept of "Key" in Relational Model

- Superkey (SK) - subset of attributes of relation which uniquely identify a tuple
  - t1[SK] ≠ t2[SK] (i.e., tuple 1's SK does not equal tuple 2's SK)
  - Superkeys for CUSTOMER relation (custno, custname, custadd, credlimit):
    - SK(Customer) = (custno, custname, custadd, credlimit)
    - SK(Customer) = (custno, custname, custadd)
    - SK(Customer) = (custno, custname)
    - SK(Customer) = (custno)
  - PRIMARY KEY - ***minimal superkey***
    - PK(Customer) = (custno)
    - PK(Orderline) = (orderno, productno)
- A relation may have more than one potential primary (candidate) key
  - EMPLOYEE(emp#,.....,taxfile#)
    - Each key ⇨ candidate key (CK)
      - » Chosen key ⇨ primary key (PK)
      - » Other keys ⇨ alternate keys (AK)

Summary of Relational Keys

- Superkey
  - Any key that uniquely identifies each entity
- Candidate key
  - Minimal superkey
- Primary key
  - Candidate key (chosen) to uniquely identify all other attributes in a given row
- Secondary key
  - Used only for data retrieval
- Composite key
  - Key composed of more than one attribute
- Key attribute
  - Any attribute that is part of a key
- Foreign key
  - Values must match a primary key in a referenced (parent) table or be null

Entity Integrity: Selecting Primary Keys

- Entity Integrity – each tuple (entity instance) must be unique
  - sets cannot have duplicate elements (relational model)
  - Relational implementation model requires
    - Primary key must be unique and not null
- Primary key is the most important characteristic of an entity
  - Single attribute or some combination of attributes
- Primary key's function is to guarantee entity integrity
- Primary keys and foreign keys work together to implement relationships (logical relationships – behind the scenes hidden from users)
  - Properly selecting primary key has ***direct*** bearing on efficiency and effectiveness

Natural Keys and Primary Keys

- Natural key is a real-world identifier used to uniquely identify real-world objects
  - Familiar to end users and forms part of their day-to-day business vocabulary
- Generally, data modeler uses natural identifier as primary key of entity being modeled
- May instead use composite primary key or surrogate key
- Surrogate key helpful when there is:
  - No natural key (real-world identifier)
  - Selected candidate key has embedded semantic contents
  - Selected candidate key is too long or cumbersome
- Do not use Surrogate key on conceptual model
- **Do not include "PK for Primary Key" on conceptual model. "**Key" should be shown instead.

TABLE 5.3 Desirable Primary Key Characteristics

PK CHARACTERISTIC	RATIONALE
Unique values	The PK must uniquely identify each entity instance. A primary key must be able to guarantee unique values. It cannot contain nulls.
Nonintelligent	The PK should not have embedded semantic meaning other than to uniquely identify each entity instance. An attribute with embedded semantic meaning is probably better used as a descriptive characteristic of the entity than as an identifier. For example, a student ID of 650973 would be preferred over *Smith, Martha L.* as a primary key identifier.
No change over time	If an attribute has semantic meaning, it might be subject to updates, which is why names do not make good primary keys. If *Vickie Smith* is the primary key, what happens if she changes her name when she gets married? If a primary key is subject to change, the foreign key values must be updated, thus adding to the database work load. Furthermore, changing a primary key value means that you are basically changing the identity of an entity. In short, the PK should be permanent and unchangeable.
Preferably single-attribute	A primary key should have the minimum number of attributes possible (irreducible). Single-attribute primary keys are desirable but not required. Single-attribute primary keys simplify the implementation of foreign keys. Having multiple-attribute primary keys can cause primary keys of related entities to grow through the possible addition of many attributes, thus adding to the database work load and making (application) coding more cumbersome.
Preferably numeric	Unique values can be better managed when they are numeric, because the database can use internal routines to implement a counter-style attribute that automatically increments values with the addition of each new row. In fact, most database systems include the ability to use special constructs, such as Autonumber in Microsoft Access, to support self-incrementing primary key attributes.
Security-compliant	The selected primary key must not be composed of any attribute(s) that might be considered a security risk or violation. For example, using a Social Security number as a PK in an EMPLOYEE table is not a good idea.

Q5 What would be your choice of Primary Key for the following data:

TABLE 5.4 Data Used to Keep Track of Events

DATE	TIME_START	TIME_END	ROOM	EVENT_NAME	PARTY_OF
6/17/2012	11:00AM	2:00PM	Allure	Burton Wedding	60
6/17/2012	11:00AM	2:00PM	Bonanza	Adams Office	12
6/17/2012	3:00PM	5:30PM	Allure	Smith Family	15
6/17/2012	3:30PM	5:30PM	Bonanza	Adams Office	12
6/18/2012	1:00PM	3:00PM	Bonanza	Boy Scouts	33
6/18/2012	11:00AM	2:00PM	Allure	March of Dimes	25
6/18/2012	11:00AM	12:30PM	Bonanza	Smith Family	12

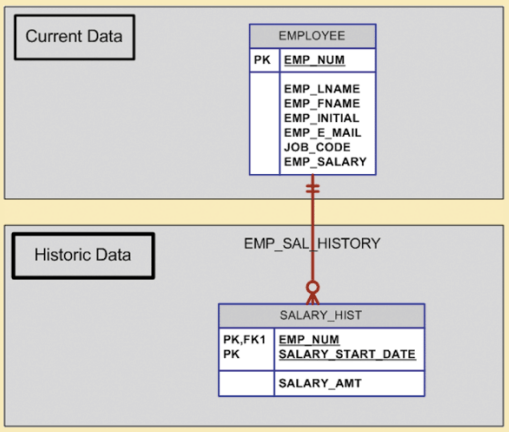
a. DATE
b. TIME_START
c. DATE and TIME_START
d. DATE, TIME_START and EVENT_NAME
e. A surrogate key EVENT_ID
f. Something else

33

Special Modeling Issues: Maintaining History of Time-Variant Data

- Normally, existing attribute values are replaced with new values without regard to previous value
- Time-variant data:
  - Values change over time
  - Must keep a history of data changes
- Keeping history of time-variant data is equivalent to having a multivalued attribute in your entity
- Must create new entity in 1:M relationships with original entity
- New entity contains new value, date of change

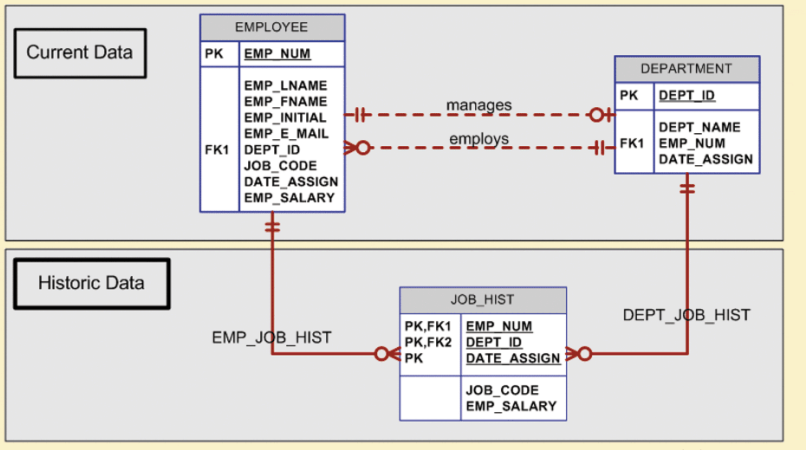
FIGURE 5.8 Maintaining salary history



SOURCE: Course Technology/Cengage Learning

35

FIGURE 5.10 Maintaining job history



SOURCE: Course Technology/Cengage Learning

Special Modeling Issues: Fan Traps

- ***Design trap*** occurs when relationship is improperly or incompletely identified
  - Represented in a way not consistent with the real world
- Most common design trap is known as ***fan trap***
- Fan trap occurs when one entity is in two 1:M relationships to other entities
  - Produces an association among other entities not expressed in the model

FIGURE 5.11 Incorrect ERD with fan trap problem

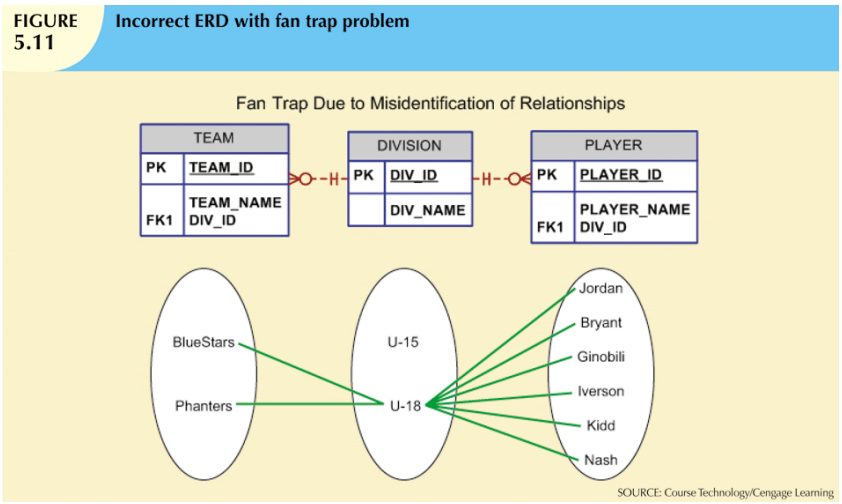
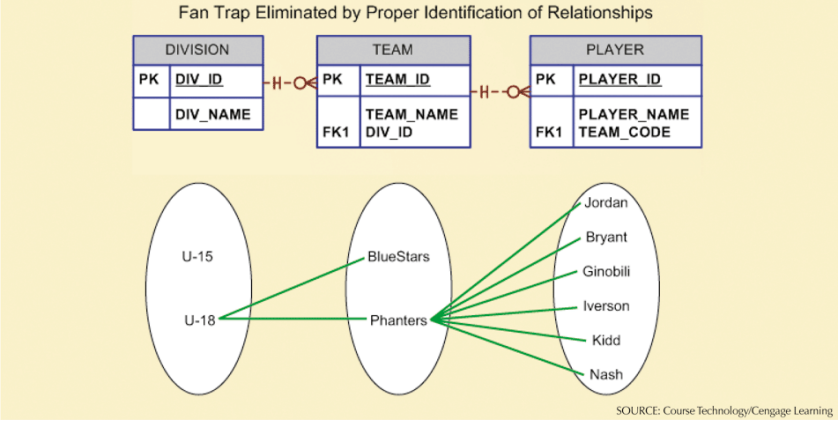


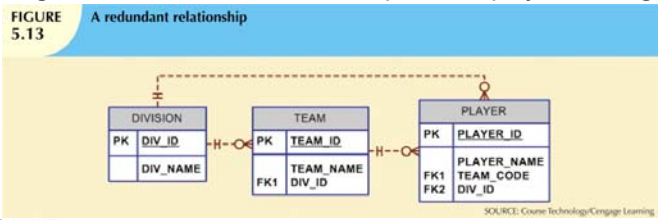
FIGURE 5.12 Corrected ERD after removal of the fan trap



Special Modeling Issues: Redundant Relationships

- Redundancy is seldom a good thing in a database environment
  - Exception is the minimal redundancy require to represent relationships
- Occurs when there are multiple relationship paths between related entities
- Main concern is that redundant relationships remain consistent across model
- Some designs use redundant relationships to simplify the design

FIGURE 5.13 A redundant relationship



Summary

- Discussed reasons why NULLs are needed and why they lead to problems such as unable to carry out certain computations or resulting in unexpected outcomes
- Discussed advantages gained by introducing supertype entity which contains common characteristics, and suptype entities, each contains their own unique characteristics.
- Discussed how to model using an Extended Entity Relationship Diagram (EERD)
- Discussed how to select appropriate primary key for an entity
- Understood the role of natural keys, composite keys and surrogate keys
- Discussed how to design a data model to deal with
  - 1:1 relationships
  - maintaining a history of time-variant data
  - fan traps and redundant relationships