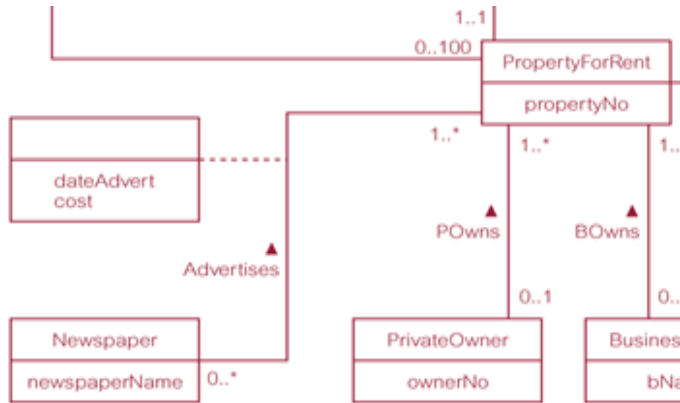


Practice Problem



Many to many relationship
Optional PropertyForRent participation
Mandatory Newspaper participation
Relationship-specific attributes

Newspaper

newspaperName

PropertyForRent

propertyNumber

Advertises

newspaperName

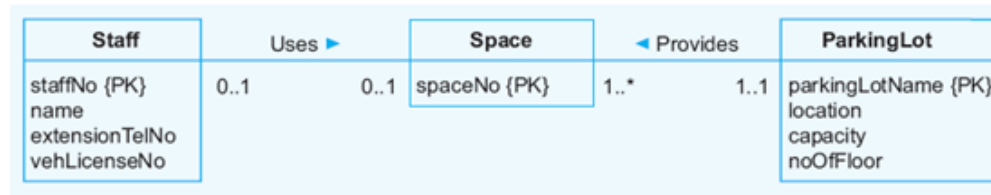
propertyNumber

dateAdvert

cost

Practice Problem

Build all the tables necessary to represent in a relational model the complete E/R model given below.



Practice Problem



Staff

<u>staffNo</u>	name	extensionTelNo	vehLicenseNo
----------------	------	----------------	--------------

ParkingLot

<u>parkingLotName</u>	location	capacity	noOfFloor
-----------------------	----------	----------	-----------

Space & Provides

<u>space</u>	parkingLotName
--------------	----------------

Uses

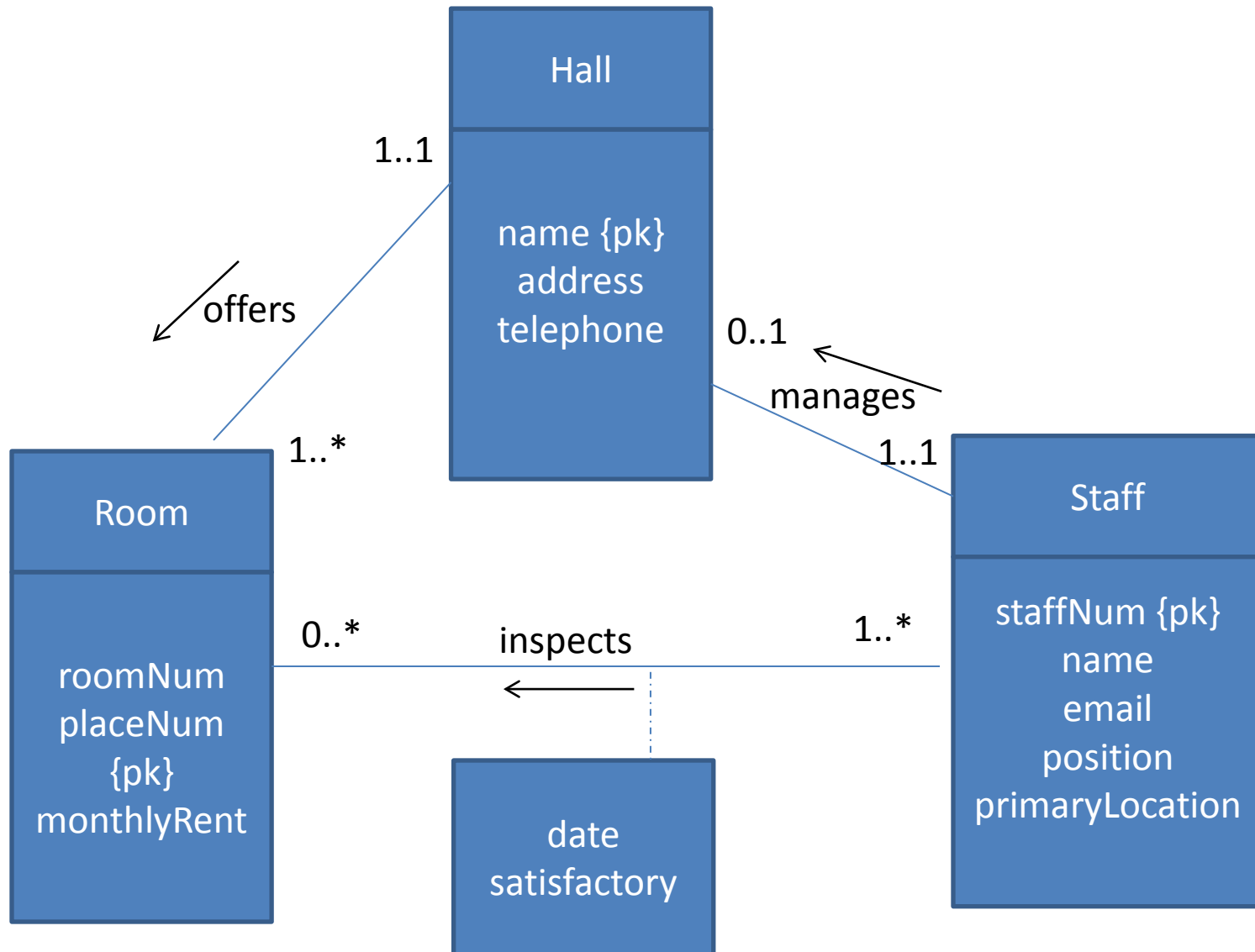
<u>staffNo</u>	<u>space</u>
----------------	--------------

Practice Problem

Build an E/R model, and an equivalent relational model for the following:

- *The Acme University Residence Office offers residence hall rooms. Each residence hall has a name, address, telephone number, and hall-manager (a member of the Residence Office staff). Only single rooms are rented. Each room has a room number, place number (unique across all rooms across the University), and monthly rent rate.*
- *Each room is inspected on a regular basis for health and sanitary reasons. The information recorded for each inspection is the staff member who carried out the inspection, the date of the inspection, and whether or not the room was found to be in satisfactory condition.*
- *Information on Residence Office staff members is maintained, including their staff number, name, email, position, and primary location (what building they work in on campus).*

Practice Problem – E/R diagram



Practice Problem – Relational Model

Hall & Manages

<u>name</u>	address	telephone	staffNum (manager)
-------------	---------	-----------	-----------------------

Room & Offers

<u>placeNum</u>	roomNum	monthlyRent	name (hall)
-----------------	---------	-------------	-------------

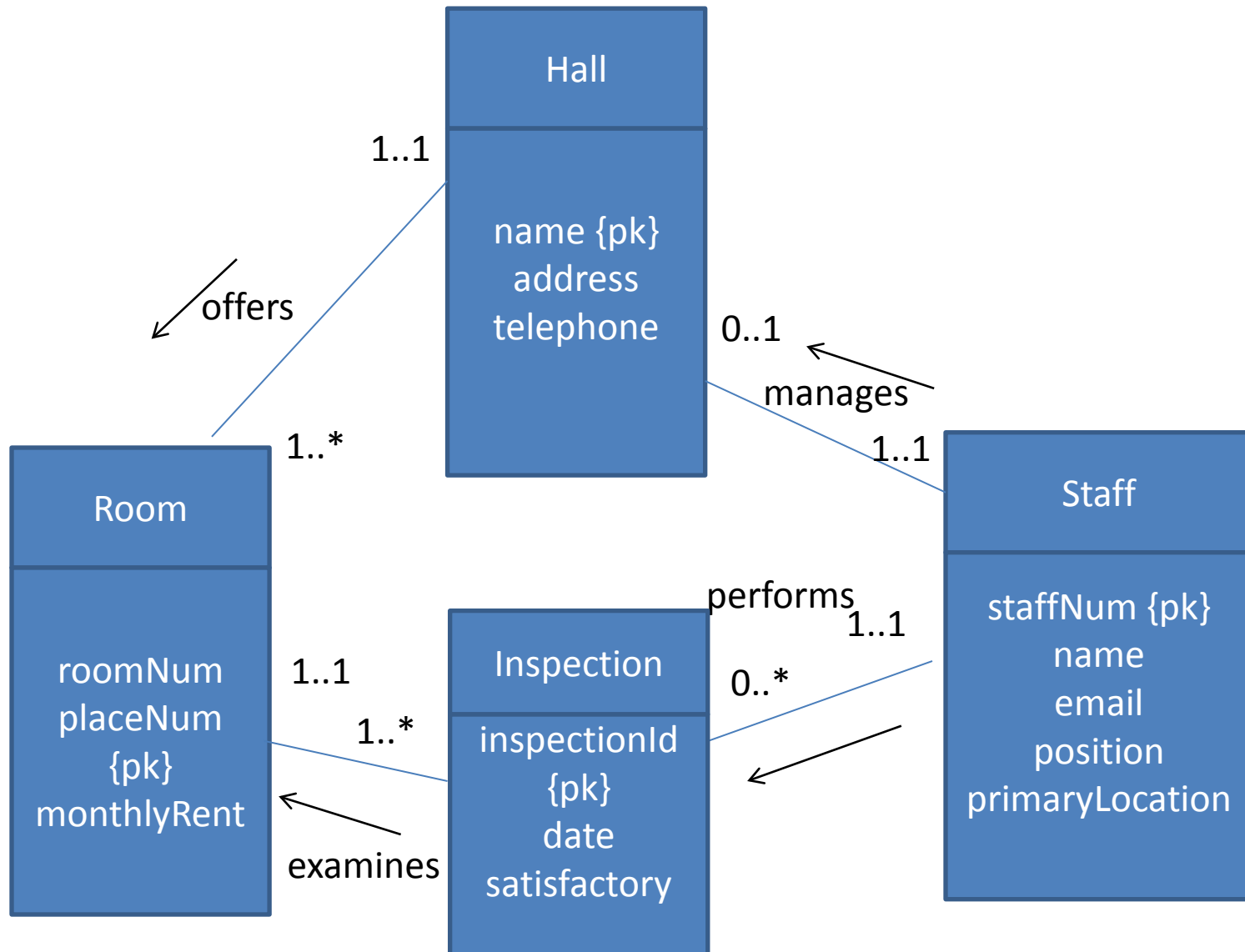
Staff

<u>staffNum</u>	name	email	position	primaryLocation
-----------------	------	-------	----------	-----------------

Inspects

<u>placeNum</u>	<u>staffNum</u>	<u>date</u>	satisfactory
-----------------	-----------------	-------------	--------------

Practice Problem – E/R diagram



Practice Problem – Relational Model

Hall & Manages

<u>name</u>	address	Telephone	staffNum (manager)
-------------	---------	-----------	-----------------------

Room & Offers

<u>placeNum</u>	roomNum	monthlyRent	name (hall)
-----------------	---------	-------------	-------------

Staff

<u>staffNum</u>	name	email	position	primaryLocation
-----------------	------	-------	----------	-----------------

Inspection & inspects & examines

<u>inspectionID</u>	date	satisfactory	staffNum (inspector)	placeNum (examined)
---------------------	------	--------------	-------------------------	------------------------

Is this the same as before?

Foreign Key

- Defined as “an attribute, or set of attributes, within one relation that matches the candidate key of some (possibly the same) relation”.
 - Exploit when modeling *relationships* in relational model
 - FK *targets* PK in *home relation*, FK in *referencing relation*

ParkingLot

<u>parkingLotName</u>	location	capacity	noOfFloor
-----------------------	----------	----------	-----------

Space & Provides

<u>space</u>	<i>parkingLotName</i>
--------------	-----------------------

Foreign Key

- *Why candidate key?*
 - Matching against an entry (a tuple) in the other relation
 - Any candidate key can identify a tuple
 - Almost always use primary key though (DBMS may require PK use)
- *Why “(possibly the same) relation”?*
 - Recursive relations
 - “Manages” is a recursive Staff-Staff relationship, usually 1:many



Foreign Keys: More Concepts

- Can be composite: sets of attributes
- Most DBMS allow the FK attributes to be mapped to new names that are more meaningful in their relation (such as a *role*)



Foreign Keys: Referential Integrity

- The value of a given instance of a foreign key must match the value of the candidate key of an instance in the target (home) relation
 - Alternatively stated: The database cannot contain any unmatched foreign keys
 - Simple statement: You can't refer to something that is not there
 - Must the opposite hold?

We will deal with NULLs in a few slides...

Integrity Constraints/Effects

- It is the DBMS's responsibility to preserve referential integrity
 - Should not allow violations of referential integrity rule in database
 - What type of changes to data could violate the rule?
 - Deletion of data from the home relation
 - Modification of PK data in the home relation
 - Insertion of bogus FK attributes in the referencing relation
 - Modification to bogus FK attributes in data in the referencing relation

Integrity Constraints/Effects

- Two common approaches to not allowing violations of referential integrity rule in database
 - RESTRICTION: Only allow operations if guarantee won't violate integrity rules
 - CASCADES: Allow operations, and then have the DBMS make necessary changes throughout the database to keep things consistent

Integrity Constraints/Effects

- Dealing with deletions (and updates) of a tuple from the home relation:
 - RESTRICTION: Only allow the tuple to be deleted (updated) if there are no foreign keys that reference the tuple
 - CASCADE: Allow the tuple to be deleted (updated), and then sweep through and delete all tuples in other relations with FK that referenced the tuple deleted (updated)
 - Note, this may need to be propagated through other relations:
R3 FK references R2, which has a FK which references R1, which is being deleted

Integrity Constraints/Effects

- If staff member is fired, what effects does removal from database have?

Staff

<u>staffNum</u>	name	email	position	primaryLocation
0001	John Doe	jd@gmail.com	HallAdviser	Luter
0002	Jane Smith	js@gmail.com	Manager	Collins
0005	Rob Van Winkle	ice@ice.com	HallAdviser	South

Inspects

<u>placeNum</u>	<u>staffNum</u>	<u>date</u>	satisfactory
L301	0002	1/30/2012	Y
L302	0001	12/20/2011	N
C503	0002	10/15/2011	N

Integrity Constraints/Effects

- Issues with deletions from Staff


Staff

<u>staffNum</u>	name	email	position	primaryLocation	
0001	John Doe	jd@gmail.com	HallAdviser	Luter	Reject/Cascade
0002	Jane Smith	js@gmail.com	Manager	Collins	Reject/Cascade
0005	Rob Van Winkle	ice@ice.com	HallAdviser	South	OK

Inspects

<u>placeNum</u>	<u>staffNum</u>	<u>date</u>	satisfactory
L301	0002	1/30/2012	Y
L302	0001	12/20/2011	N
C503	0002	10/15/2011	N

Integrity Constraint/Effects



<u>staffNum</u>	name	email	position	<i>manager</i>
0001	Calvin Broadus	snoop@gmail.com	Teller	0002
0002	Andre Young	dre@gmail.com	TellerManager	0003
0003	Warren Griffin	warren@gmail.com	Supervisor	0005
0004	Stan Burrell	hammer@gmail.com	TellerManager	0003
....				

If we fire Andrew Young (delete 0002),
do we delete 0001 as well with a cascading delete?

(Not clear cut → may assign 0001 to have a new manager first).

Another example: Orders, ShipmentsOfOrders (delete an order?)

Integrity Constraints/Effects

- Dealing with “bogus” insertions/modifications to referencing table:
 - DBMS should check that the value be inserted/updated for the foreign key matches a primary key instance of the home relation; if not, reject
 - DBMS should not change home relation!

NULL

- Book definition: Used to represent a value for an attribute that is not currently known or not applicable for a given tuple”.
 - Doesn't exist/Absence of a value/Not applicable
 - Examples:
 - The president of the company (a staff member) does not have a manager
 - Property viewer did not make a comment after viewing
 - New hire has not been assigned branch yet

Integrity Rules with NULL

- Entity integrity:
 - Primary key can't be NULL in any base relation
 - Primary keys are unique identifiers - they essentially provide the identity to the tuple
 - NULL for a PK means “no identity”
 - A composite primary key can't be partially NULL
 - If so, suggests the PK can be minimized more, so it wasn't actually a reasonable PK

Problems with NULLS

- There are some concerns with the uses of NULLs in databases
 - Introduce entity integrity anomalies
 - Violate fundamental predicate logic calculus that relational model operations built on

Entity Integrity Anomalies

- Definition appears to be limited to base relations
 - Base relation: a relation defined *by us*, not derived from an operation on a relation
 - A projection is an operation that transforms a relation to another relation, by selecting a subset of attributes (and also removing repeated rows to preserve ‘relation as a set’ property)
 - Assume we project to one attribute and there is a tuple with a NULL set as that attribute. That one attribute must be the PK of the newly formed relation.
 - Allow a NULL as PK?
 - Throw away the tuple with the NULL in it?

Entity Integrity Anomalies

- Definition appears to be limited to primary key, not candidate keys
 - If a candidate key attribute could have a NULL, was it really a candidate key to begin with?
 - Candidate key – minimal unique identifier

Predicate Logic Problems

- Introduction of NULLs requires more than true/false in analyzing relations
 - Consider `tupleA.attributeA == tupleB.attributeA`
 - Normally, `==` results in either true or false;
 - Similarly: `>`, `<`, `>=`, `<=`, `!=`
 - What if values are NULL and NULL
 - Is `NULL == NULL`, is `NULL > NULL`, is `NULL < NULL`?
 - Can't answer true or false – have to answer “Unknown”

Integrity Rules with NULL

- Referential integrity:
 - If NULLs allowed, a foreign key reference, if it does not reference a valid PK, should be wholly NULL
 - Wholly null: if the FK is a composite key, all attributes of the FK should be null
 - Instead of cascading deletes of whole tuples, we could perform cascading updates, converting FK to NULL when PK that the FK refers to is deleted

Food for Thought: NULL

- Can we design around NULLs?
 - Can we build a relational model such that NULL entries are not required?
 - Can we build a relational model such that NULL entries are minimized?
- Most DBMSs will support specifying that attributes can't be NULL
 - Explicitly or implicitly (i.e. if specify as PK)

Database Creation

- At this point, we have seen much of the material needed to be able to specify a database schema we would be interested in using
 - Attributes
 - Primary keys
 - Foreign keys
 - Integrity constraint options (RESTRICT, CASCADE)
- We haven't yet dealt with the tuples themselves yet

Database Languages (Chapter 2)

- Structure-oriented:
 - Data definition language (DDL): “A language that allows ... the description and naming of the entities, attributes, and relationships required for the application, with any associated integrity and security constraints.”
- Data-oriented:
 - Data manipulation language (DM): “A language that provides a set of operations to support the basic data manipulation operations on the data held in the database.”
 - Four essential operations: Insert, Modify, Delete, Retrieve

Database Languages (Chapter 2)

- SQL (Structured Query Language) actually acts as both a DDL and a DML
 - DDL: Chapter 7
 - DML: Chapter 6

Creating Databases: Defining Tables

Key DDL Functionality To Start:

- Create database (named collection of relations/tables)
- Create domains for attributes
- Create named tables, specifying
 - Attributes and associated domains
 - Primary keys
 - Foreign keys and references
 - Integrity constraints decisions
- Can also remove (“Drop”) databases, tables, domains
- Can modify (“Alter”) tables and domains

Database Virtual Machine

- Will be provided access to a virtual machine supporting MySQL next week
- Has a private WFU address
 - Only available from on-campus or using VPN (think about what that means for your work/study habits)
- Ubuntu Linux, terminal access
 - Let me know if uncomfortable in Linux