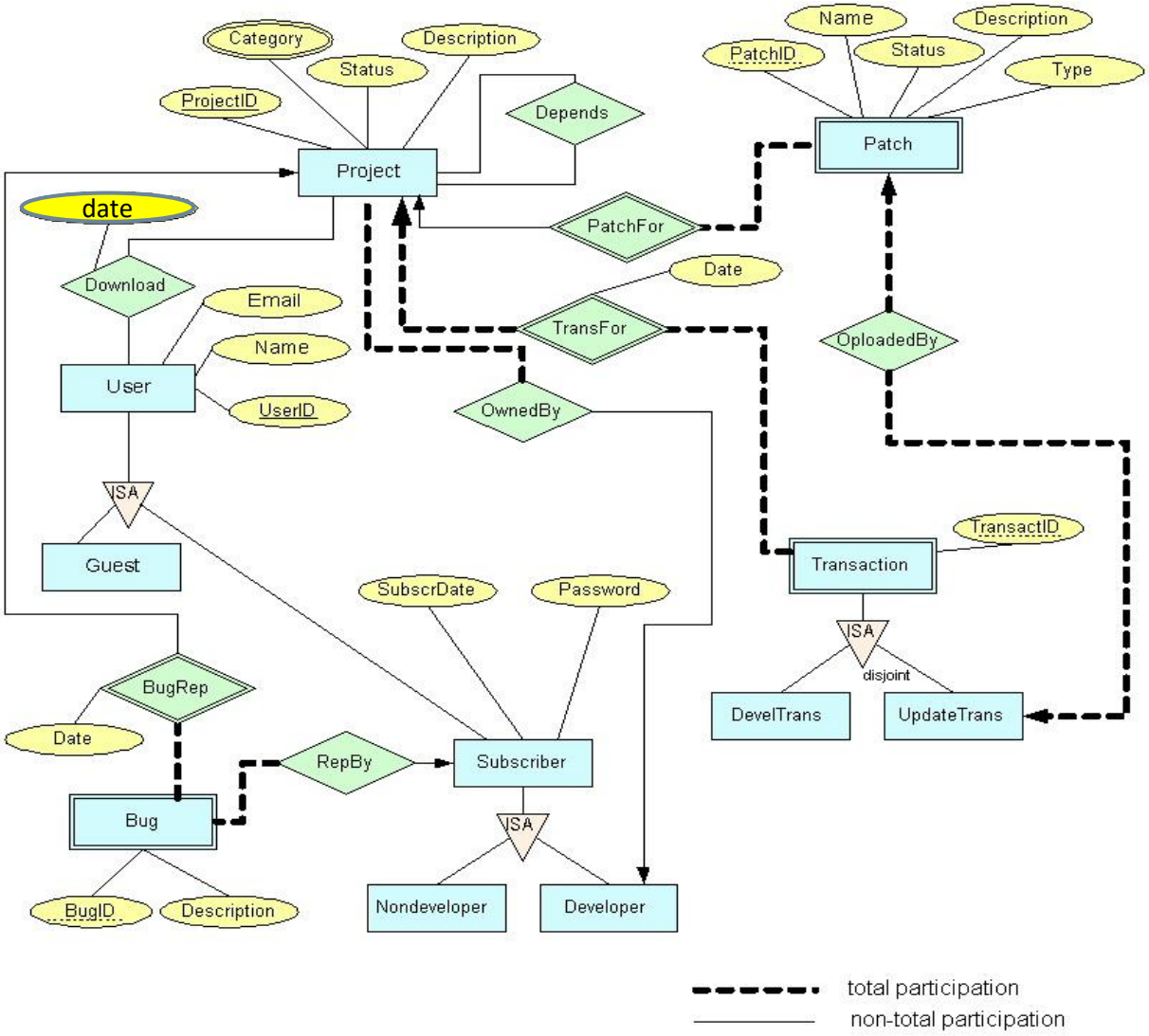
# SE 4DB3 TUTORIAL 2: REDUCTION TO RELATIONAL SCHEMAS

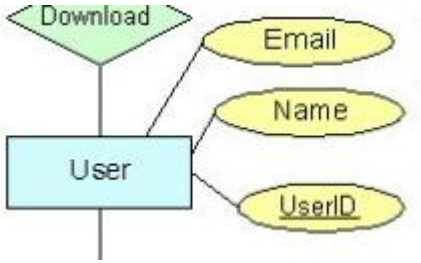
Jan 29, 2016

# Sample ER diagram



# Representing of Strong Entity Sets

- Let **E** be a strong entity set with descriptive attributes $a_1, a_2, \ldots, a_n$.
  - We represent this entity by a schema called **E** with **n** distinct attributes.
  - The primary key of the entity set serves as the **primary key** of the resulting schema.

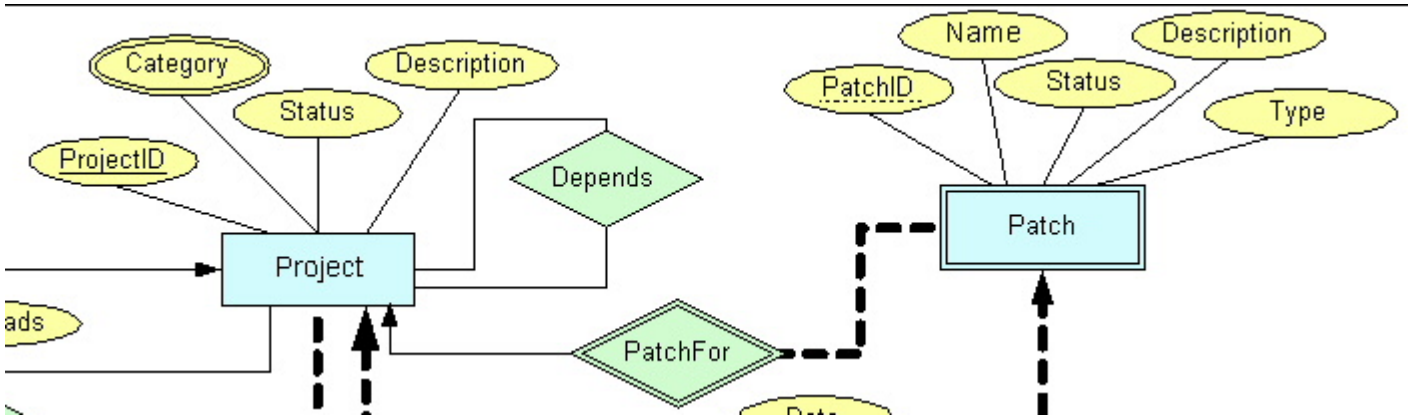



USER(userid, name, email)
PK=userid

# Representation of Weak Entity Sets

- Let **A** be a weak entity set with attributes $a_1, a_2, \ldots, a_m$. Let **B** be the strong entity set on which **A** depends. Let the **primary key** of **B** consist of attributes $b_1, b_2, \ldots, b_n$.
  - We represent the entity set **A** by a relation schema called **A** with one attribute for each member of the set: $\{a_1, a_2, \ldots, a_m\} \cup \{b_1, b_2, \ldots, b_n\}$
  - The combination of the primary key of the strong entity set and the discriminator of the weak entity set serves as **the primary key** of the schema.

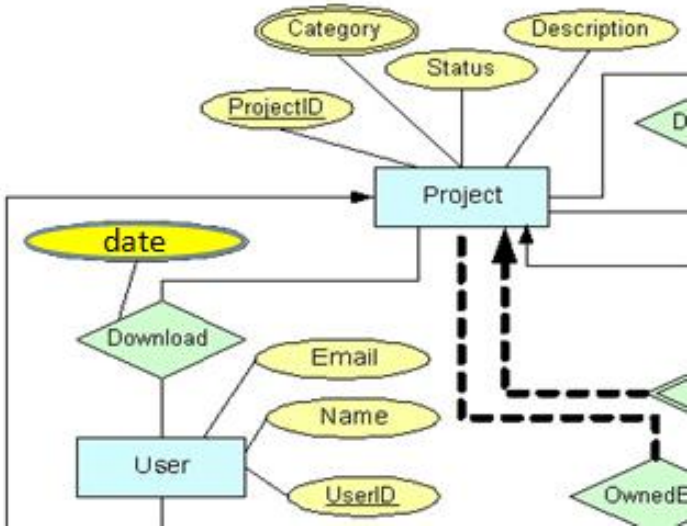
# Representation of Weak Entity Sets




PATCH(projectid, patchid, name, status, description, type)
PK=(projectid, patchid)
FK(PROJECT)=property_id

# Representation of Relationship Sets

- A **many-to-many** relationship set is represented as a schema with attributes for the primary keys of the two participating entity sets, and any descriptive attributes of the relationship set.




Schemas derived from entity sets:

PROJECT (…)

USER(…)

Schema derived from relationship sets:

DOWNLOAD(projectid, userid, date)

PK=(projectid, userid, date)

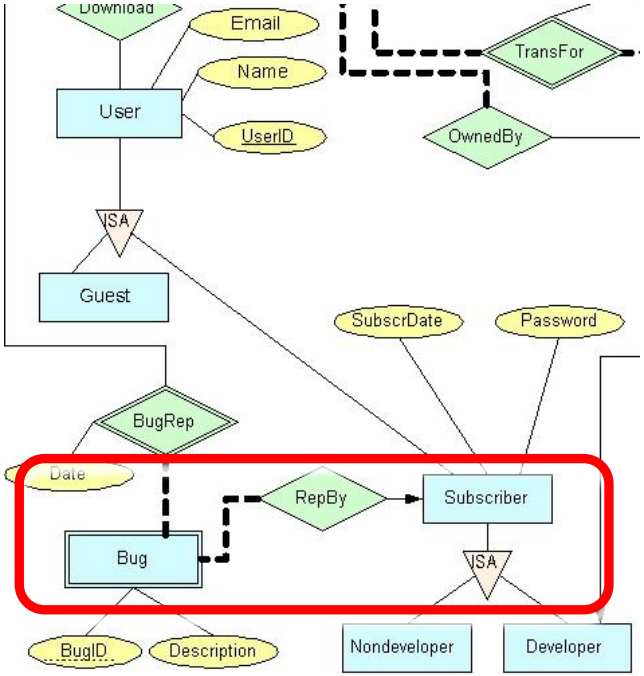
NOTE: same user may download same project on different dates.

FK(PROJECT)=property_id

FK(USER)=userid

# Redundancy of Schemas

- **Many-to-one** and **one-to-many** relationship sets that are **total** on the many-side can be represented by adding an extra attribute to the "many" side, containing the **primary key** of the "one" side.




Schemas derived from entity sets:

SUBSCRIBER(..)

BUG(projectid, date, bugid, description, userid)

NOTE: projectid and date because of weak entity set; userid because of many-to-one relationship.

PK=(projectid, bugid)

FK(PROJECT)=property_id

FK(USER)=userid

No schema derived from relationship set

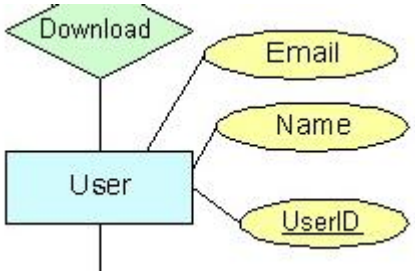
Correction of PK

# Redundancy of Schemas

- We can combine schemas even if the participation is **partial**, by using **null** values.
  - i.e. we could store null values for the userid attribute for BUG that have no associated SUBSCRIBER.
- For **one-to-one** relationship sets, **either** side can be chosen to act as the "many" side.
  - That is, extra attribute can be added to either of the tables corresponding to the two entity sets.

# Composite Attributes

- We handle composite attributes by creating a separate attribute for each of the component attributes; we do not create a separate attribute for the composite attribute itself.

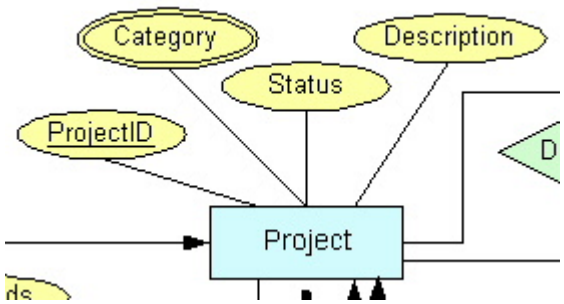



Suppose that
Name=last name + middle name + first name

USER(userid, last name, middle name, first name, email)
PK=userid

# Multivalued Attributes

- For a **multivalued attribute M**, we create a relation schema **R** with an attribute **A** that corresponds to **M** and attributes corresponding to the primary key of the entity set or relationship set of which **M** is an attribute.



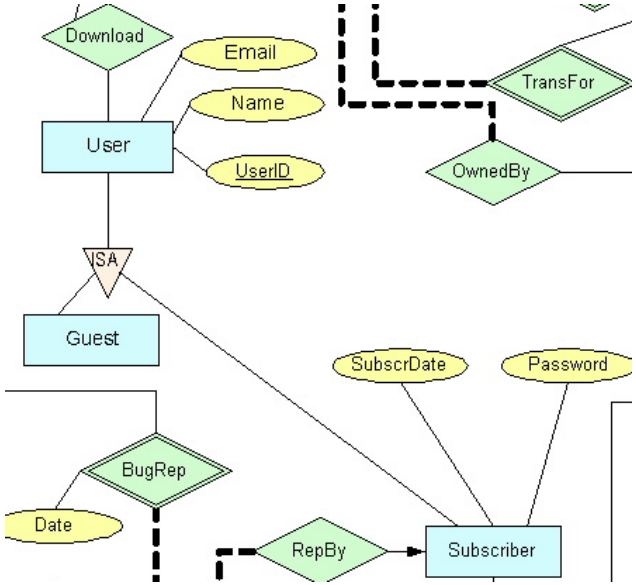

PROJECT(projectid, status, description)
PK=projectid

CATEGORY(projectid, categories)
PK=(projectid, categories)
FK(PROJECT)=projectid

Correction of PK for CATEGORY

# Representation of Specialization

- Method 1:
  - Form a schema for the **higher-level** entity
  - Form a schema for each **lower-level** entity set, include **primary key** of higher-level entity set and local attributes.



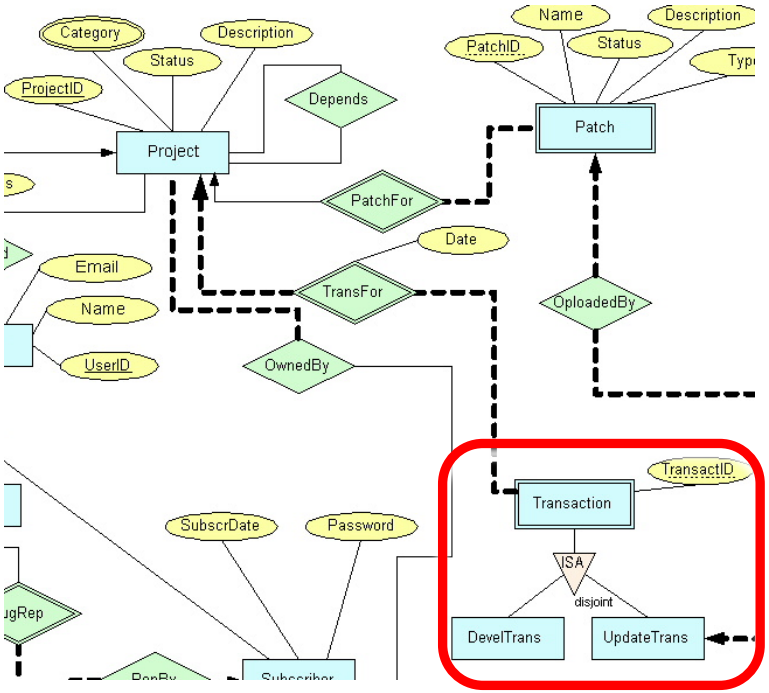

USER(userid, email, name)
PK=userid

GUEST(userid)
PK=userid
FK(USER)=userid

SUBSCRIBER(userid, subscrdate, password)
PK=userid
FK(USER)=userid

# Representation of Specialization

- Method 2:
  - If specialization is disjoint and complete
  - Form a schema for entity set with all local and inherited attributes.




TRANSACTION(projectid, date,transactid)
PK=(projectid, transactid)
FK(PROJECT)=projectid

DEVELTRANS(projectid, date, transactid)
PK=(projectid, transactid)
FK(TRANSACTION)=(projectid, transactid)

UPDATETRANS(projectid, date, transactid)
PK=(projectid, transactid)
FK(TRANSACTION)=(projectid, transactid)

Correction: Add TRANSCATION because FK constraint; correct FK for lower-level entity.