Veritabanı Yönetim Sistemleri

(Veritabanı Tasarımı)
Genişletilmiş Varlık İlişki Modeli

Konular

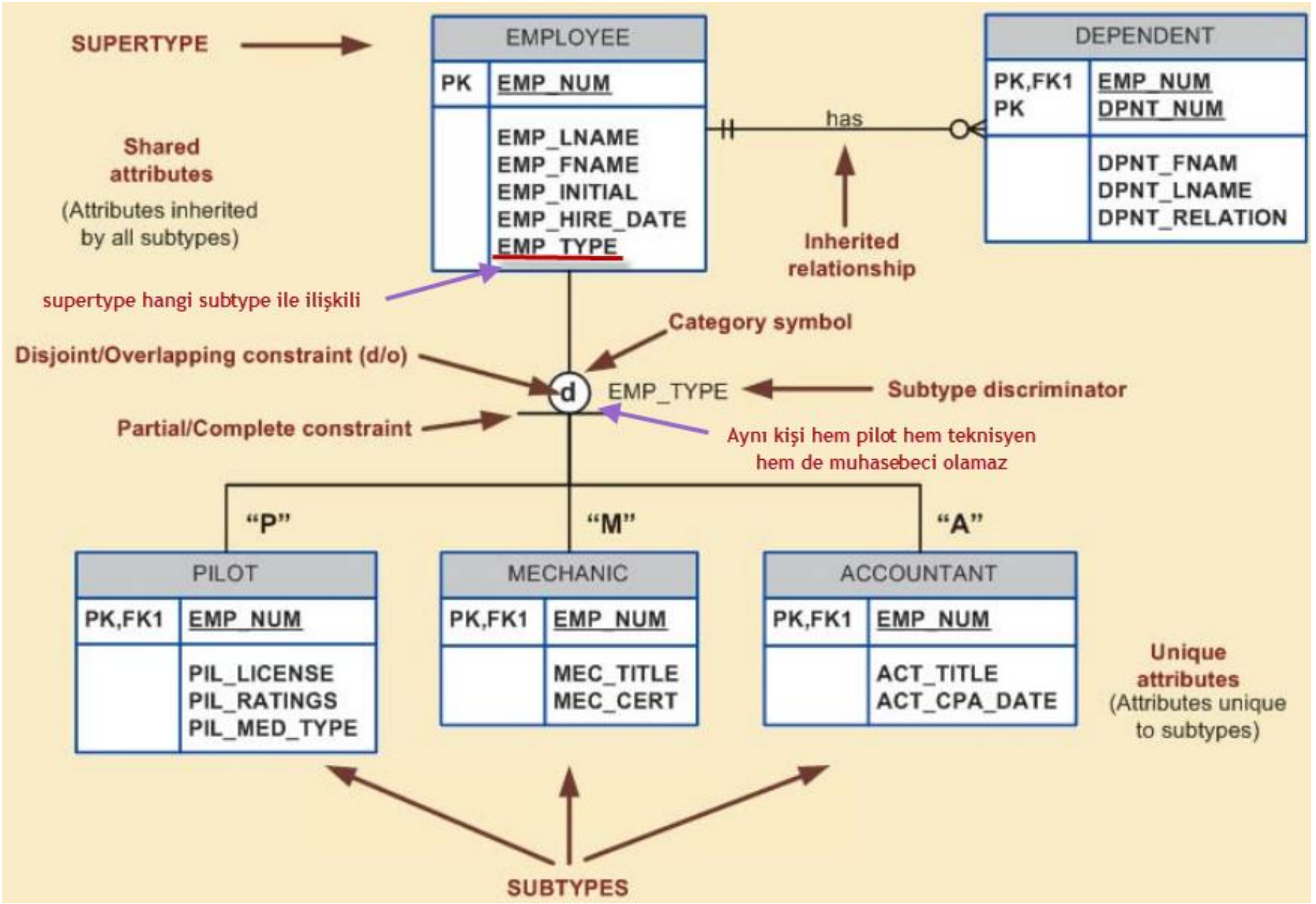
- ✓ Genelleme
- ✓ Kümeleme
- ✓ Birincil Anahtar Özellikleri
- ✓ Surrogate(Vekil, yerini tutucu) Birincil Anahtar
- ✓ Özet
- ✓ Kaynaklar

Genelleme (Kalıtım)

- ✓ Bir veritabanı içerisinde benzer özelliklere sahip varlıklar için ortak alanları içeren temel bir varlık oluşturmak ve diğer varlıkları bu temel varlıktan türetmek; daha hızlı tasarım , değişikliklerin kolay yapılabilmesi, anlaşılabilirliğin artması gibi avantajları (NYP paradigmasındaki kalıtımın sağladığı avantajların çoğunun sağlanması) beraberinde getirir.

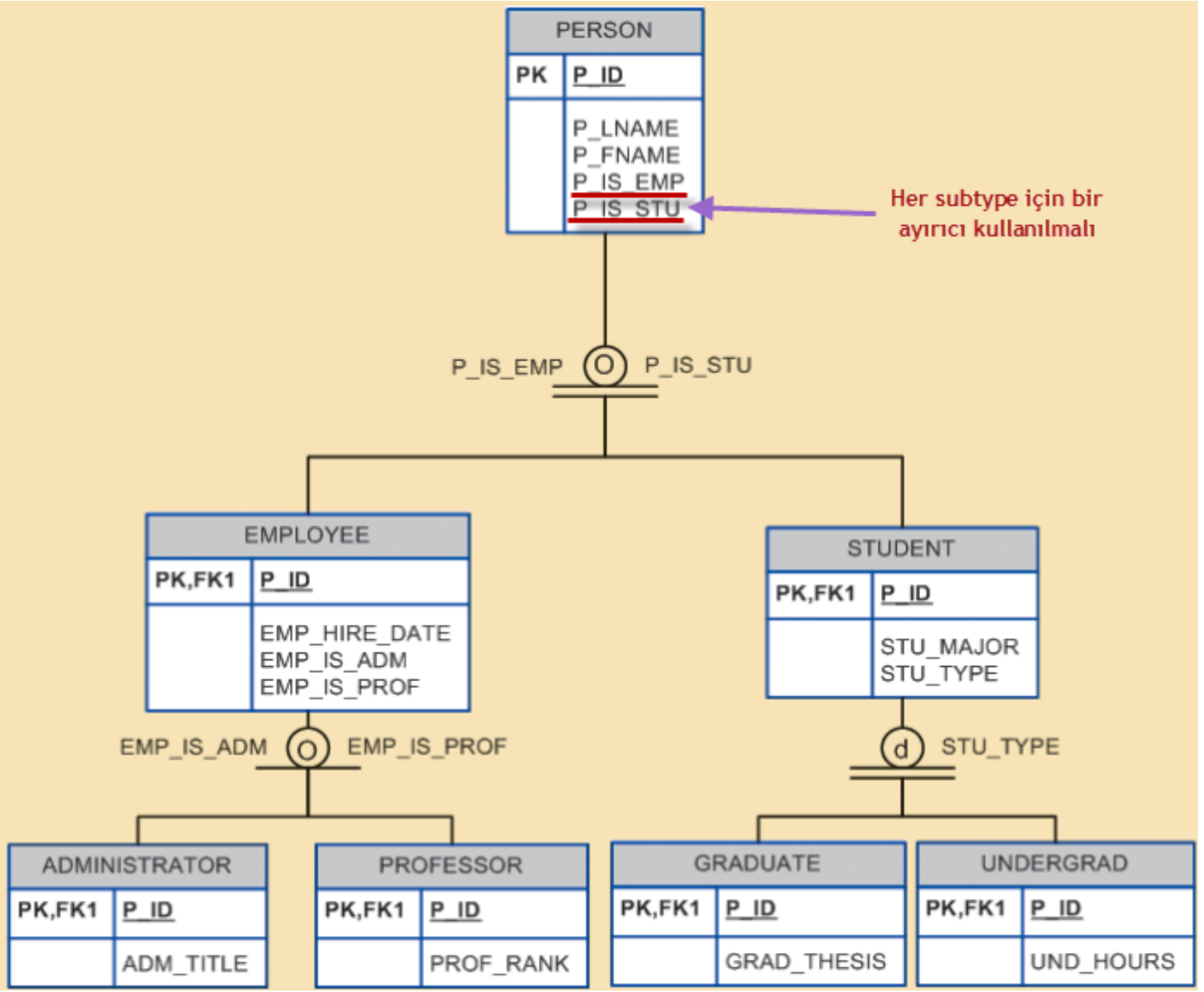
- ✓ **O (Overlap):** Aynı anda birden fazla çocuk varlık olabilir. (Hem personel hem öğrenci)
- ✓ **D (Disjoint):** Aynı anda sadece bir çocuk varlık olabilir. (Ya Lisans Öğrencisi ya da Yüksek Lisans Öğrencisi)
- ✓ **Partial completeness** (Tek çizgi olduğunda) **:** supertype kayıtlar subtype kayıtlar olmadan da mevcut olabilir.
- ✓ **Total completeness** (Çift çizgi olduğunda) **:** her supertype mutlaka en az bir subtype ın üyesi olmalı

Genelleme



Carlos Coronel, Steven Morris, and Peter Rob, Database Systems: Design, Implementation, and Management

Genelleme

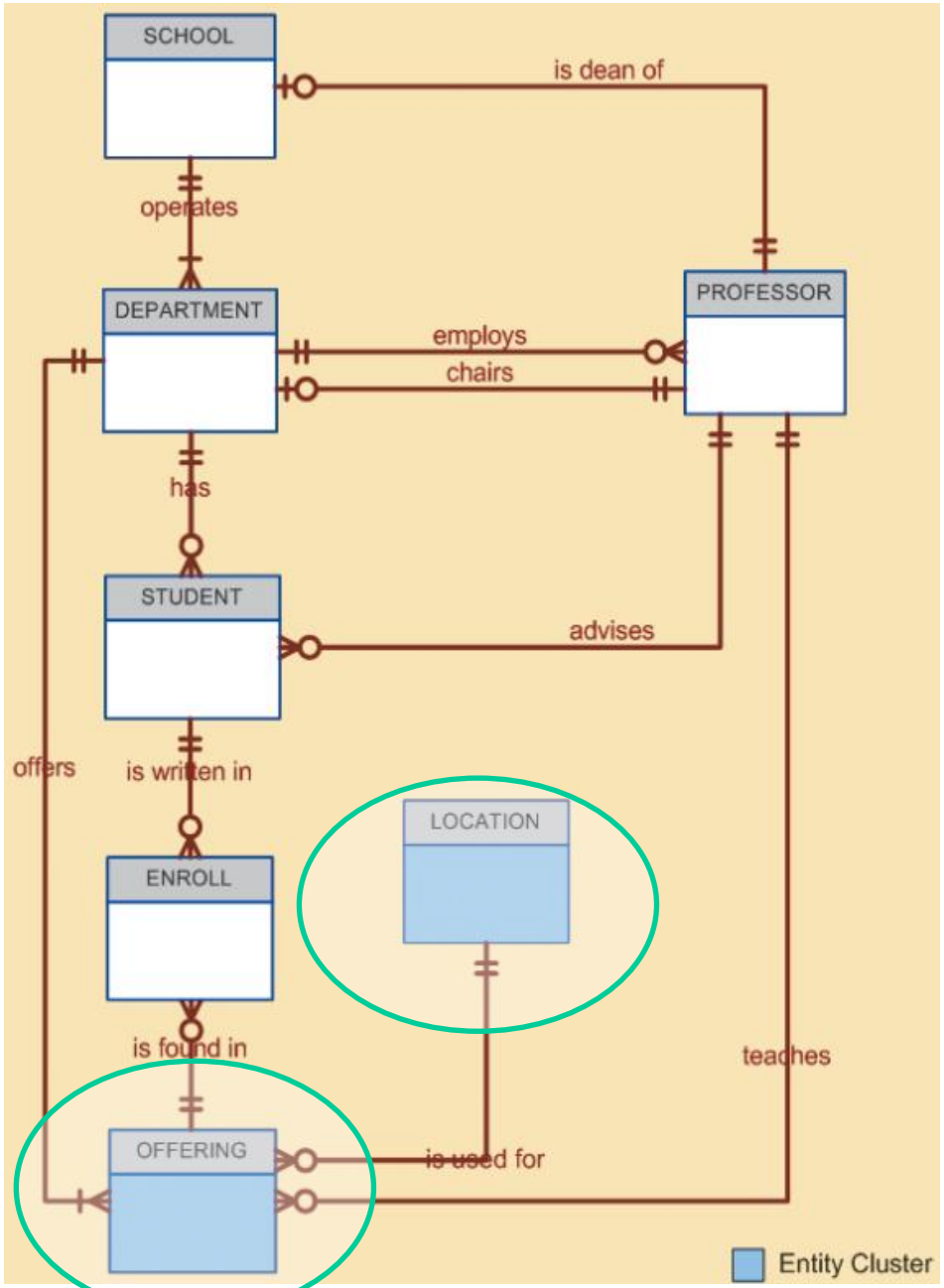


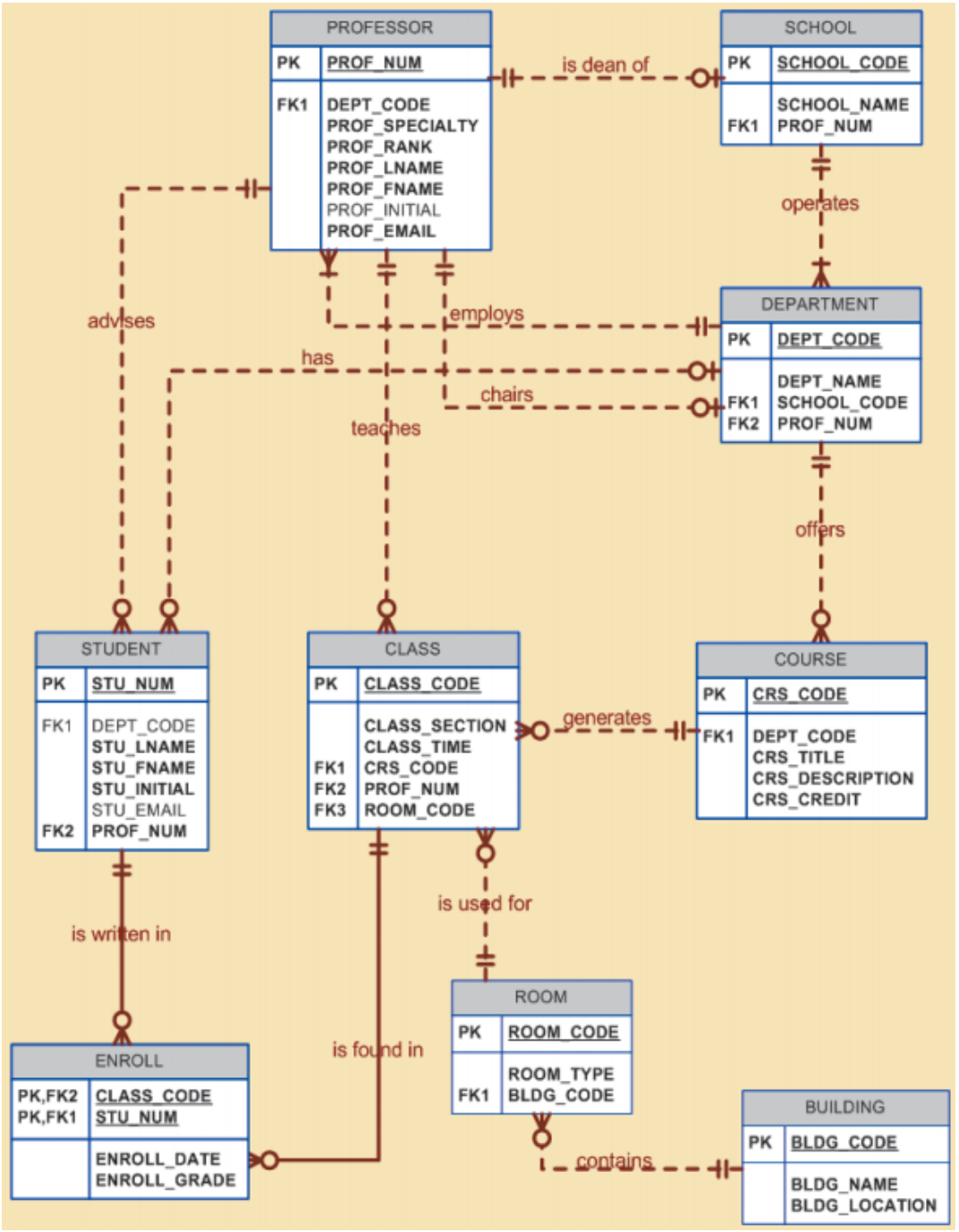
Carlos Coronel, Steven Morris, and Peter Rob, Database Systems: Design, Implementation, and Management

Kümeleme

- ✓ **Çok sayıdaki varlık ve ilişkileri Vİ şemalarını basitleştirmek ve okunabilirliğini artırmak için kullanılan sanal varlığa varlık kümesi denir.**

Kümeleme


SCHOOL
is dean of
operates
DEPARTMENT
employs
chairs
PROFESSOR
has
STUDENT
advises
offers
is written in
ENROLL
LOCATION
is found in
teaches
OFFERING
is used for
Entity Cluster


PROFESSOR
PK PROF_NUM
FK1 DEPT_CODE
PROF_SPECIALTY
PROF_RANK
PROF_LNAME
PROF_FNAME
PROF_INITIAL
PROF_EMAIL
is dean of
SCHOOL
PK SCHOOL_CODE
SCHOOL_NAME
FK1 PROF_NUM
operates
DEPARTMENT
PK DEPT_CODE
DEPT_NAME
FK1 SCHOOL_CODE
FK2 PROF_NUM
advises
employs
has
chairs
teaches
offers
STUDENT
PK STU_NUM
FK1 DEPT_CODE
STU_LNAME
STU_FNAME
STU_INITIAL
STU_EMAIL
FK2 PROF_NUM
CLASS
PK CLASS_CODE
CLASS_SECTION
CLASS_TIME
FK1 CRS_CODE
FK2 PROF_NUM
FK3 ROOM_CODE
generates
COURSE
PK CRS_CODE
FK1 DEPT_CODE
CRS_TITLE
CRS_DESCRIPTION
CRS_CREDIT
is written in
is used for
ENROLL
PK,FK2 CLASS_CODE
PK,FK1 STU_NUM
ENROLL_DATE
ENROLL_GRADE
is found in
ROOM
PK ROOM_CODE
ROOM_TYPE
FK1 BLDG_CODE
contains
BUILDING
PK BLDG_CODE
BLDG_NAME
BLDG_LOCATION

Birincil Anahtar Özellikleri

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. This is why names do not make good primary keys. If you have Vickie Smith as 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.

Carlos Coronel, Steven Morris, and Peter Rob, Database Systems: Design, Implementation, and Management

Surrogate(Vekil, yerini tutucu) Birincil Anahtar

DATE	TIME_START	TIME_END	ROOM	EVENT_NAME	PARTY_OF
6/17/2010	11:00AM	2:00PM	Allure	Burton Wedding	60
6/17/2010	11:00AM	2:00PM	Bonanza	Adams Office	12

(**DATE**, **TIME_START**, **ROOM**) or (**DATE**, **TIME_END**, **ROOM**)

Assume you select the composite primary key (**DATE**, **TIME_START**, **ROOM**) for the EVENT entity. Next, you determine that one EVENT may use many RESOURCEs (such as tables, projectors, PCs, and stands), and that the same RESOURCE may be used for many EVENTs. The RESOURCE entity would be represented by the following attributes:

RESOURCE (**RSC_ID**, RSC_DESCRIPTION, RSC_TYPE, RSC_QTY, RSC_PRICE)

Given the business rules, the M:N relationship between RESOURCE and EVENT would be represented via the EVNTRSC composite entity with a composite primary key as follows:

EVNTRSC (**DATE, TIME_START, ROOM, RSC_ID,** QTY_USED)

You now have a lengthy four-attribute composite primary key. What would happen if the EVNTRSC entity's primary key were inherited by another existence-dependent entity? At this point, you can see that the composite primary key could make the implementation of the database and program coding unnecessarily complex.

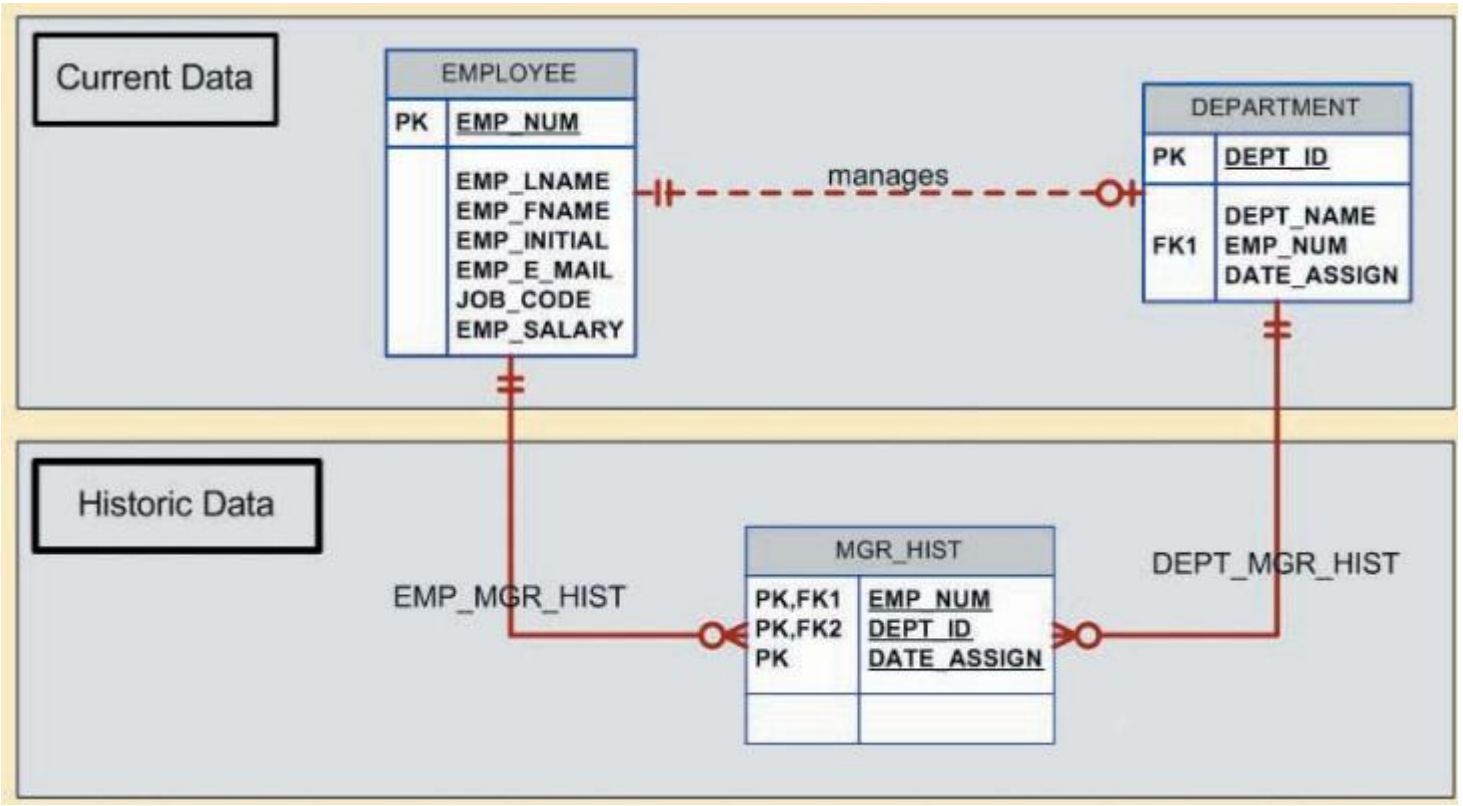
As a data modeler, you probably noticed that the EVENT entity's selected primary key might not fare well, given the primary key guidelines in Table 5.3. In this case, the EVENT entity's selected primary key contains embedded semantic information and is formed by a combination of date, time, and text data columns. In addition, the selected primary key would cause lengthy primary keys for existence-dependent entities. The preferred alternative is to use a numeric single-attribute surrogate primary key.

Surrogate primary keys are accepted practice in today's complex data environments. They are especially helpful when there is no natural key, when the selected candidate key has embedded semantic contents, or when the selected candidate key is too long or cumbersome. However, there is a trade-off: if you use a surrogate key, you must ensure that the candidate key of the entity in question performs properly through the use of "unique index" and "not null" constraints.

Carlos Coronel, Steven Morris, and Peter Rob, Database Systems: Design, Implementation, and Management

Tasarım Sırasında Karşılaşılan Sorunlar

A One-to-One (1:1) Relationship:
An EMPLOYEE manages zero or one DEPARTMENT;
each DEPARTMENT is managed by one EMPLOYEE.

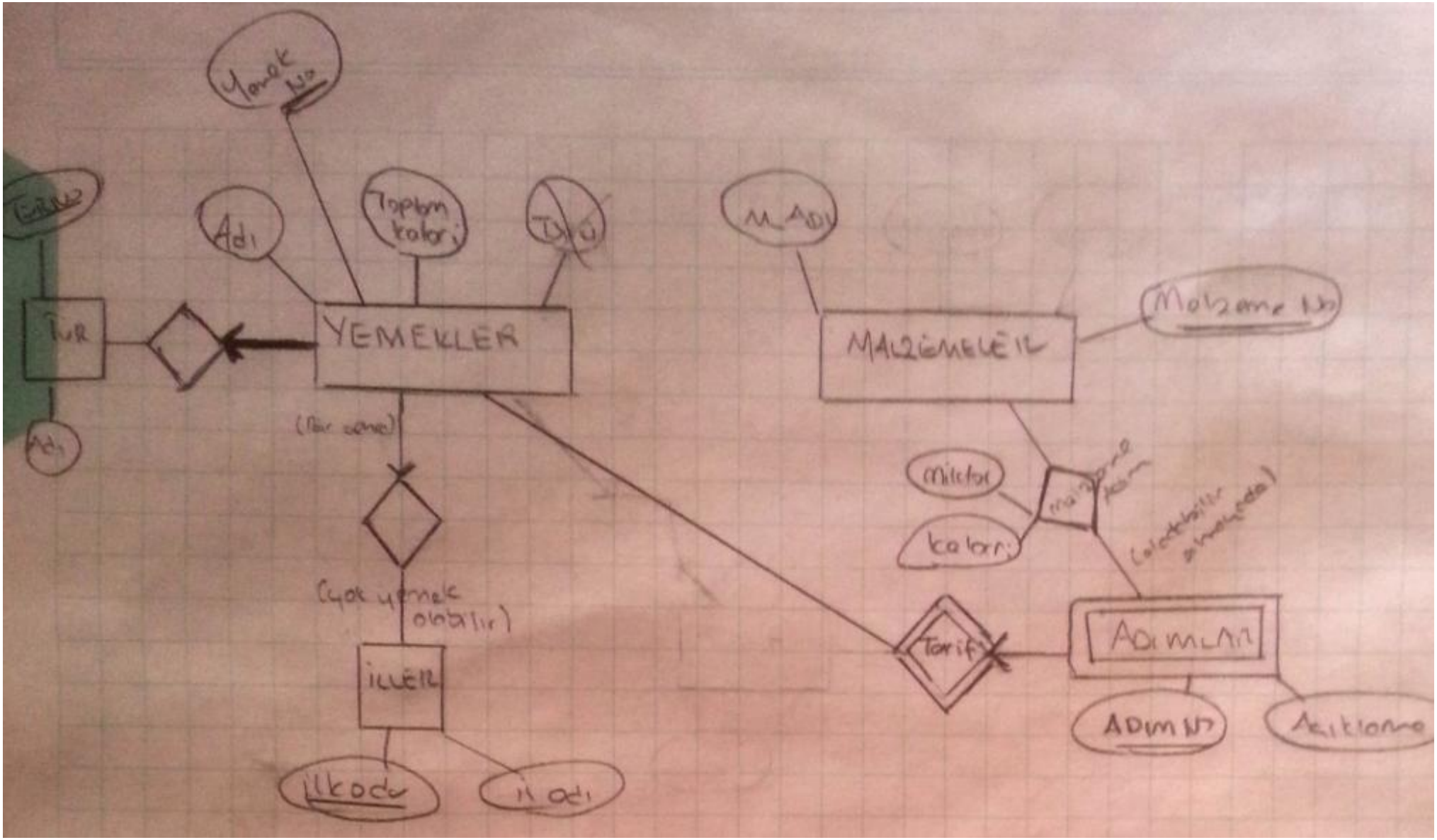


Carlos Coronel, Steven Morris, and Peter Rob, Database Systems: Design, Implementation, and Management

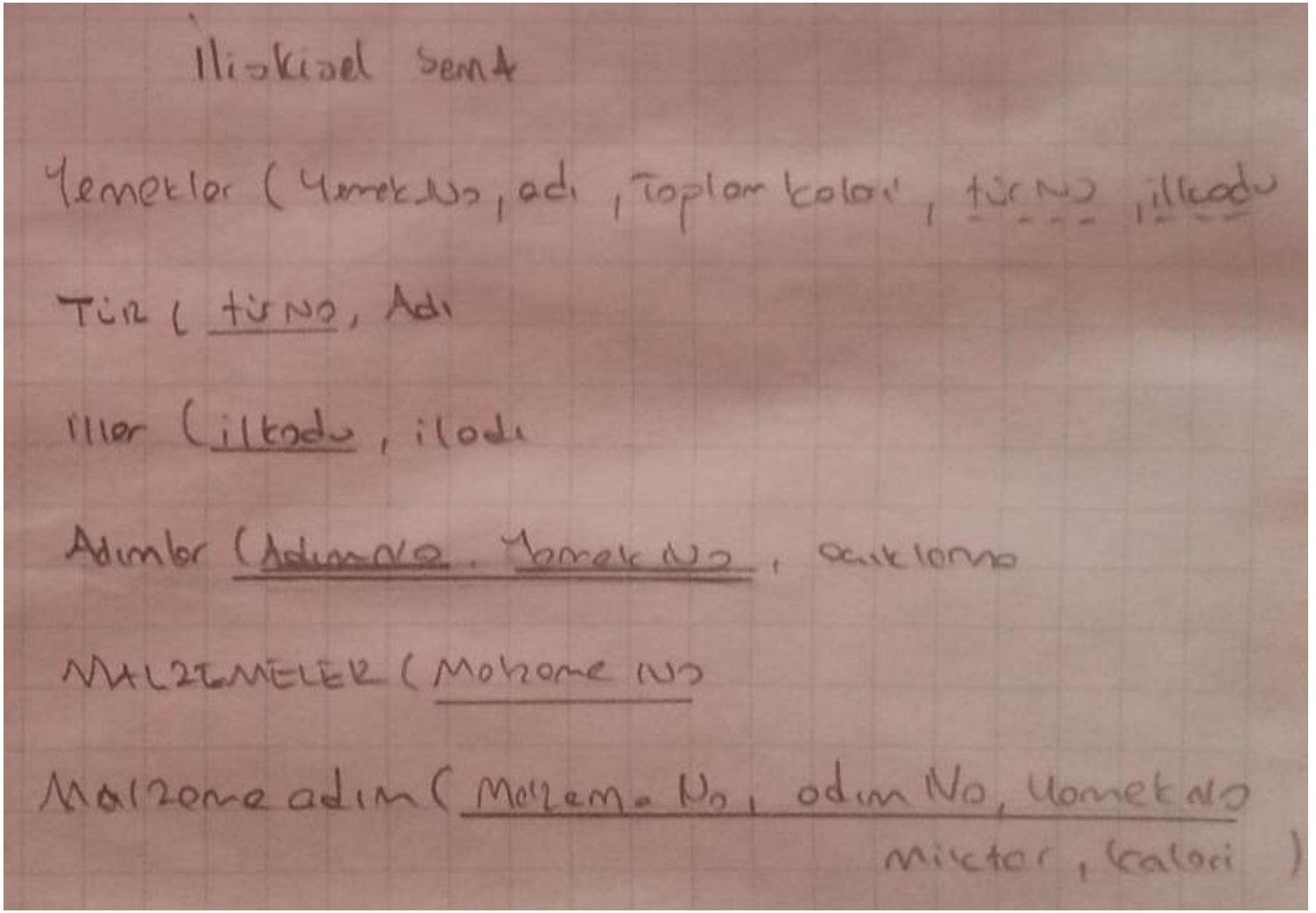
Örnek Uygulama – Yemek Tarifleri

Yemek tariflerini saklayacak bir veritabanı tasarlayınız. Her yemeğin adı, hangi şehire ait olduğu, toplam kalorisi, ve türü tutulmalıdır. Toplam kalori malzemelerin kalorileri toplamından hesaplanacaktır. Yemek türü 'ana yemek', 'tatlı', 'çorba' olabilir. Her yemek için yemeğin malzeme listesi de tutulacaktır. Bir yemeğin her malzemesi için malzeme adı, malzeme miktarı, kalorisi tutulacaktır. Her yemek için yemeğin tarifi de tutulacaktır. Yemek tarifi adımlardan oluşacaktır. Adımlar 1,2,3,.. diye numaralandırılacaktır. Her adım için, kaçıncı adım olduğu, o adımda ne yapılacağı, o adımda hangi malzemelerin kullanılacağı tutulacaktır.

Örnek Uygulama - Yemek Tarifleri



Örnek Uygulama - Yemek Tarifleri



Kaynaklar

- ✓ Carlos Coronel, Steven Morris, and Peter Rob, Database Systems: Design, Implementation, and Management, Cengage Learning.
- ✓ Raghu Ramakrishnan, Johannes Gehrke, Database Management Systems, Mc Graw Hill