سوال دوم)

قست الف) اگر دو یا چند جدول از طریق foreign key ها به هم متصل شده باشند، می‌توان با استفاده از دستور join ، به عنوان مثالی اطلاعاتی از جدول دوم که در جدول اول نیامده است را نیز مشاهده کرد. حال اگر جدول سومی نیز وجود داشته باشد که محتوی یک foreign key از جدول اول باشد، می‌توان این اطلاعات اضافی را هم به جدول مجازی که ایجاد می‌شود و در دیتا بیس ما ذخیره نمی‌شود اضافه کرد. برای درک بهتر این مفهوم می‌توان به طور کلی مفهوم foreign key را برای لحظه‌ای فراموش کرد و شکل ساده‌ای که در لینک آموزشی آمده بود را بررسی کرد.

```
Select
A,
Fruit_a,
B,
Fruit_b
From
Basket_a
Inner join basket_b
On fruit_a = fruit_b;
```

برای درک بهتر دستور را از پایین می‌خوانیم، شرط ما این است که `fruit_a` با `fruit_b` برابر باشد. در صورتی که این شرط برآورده شود، اطلاعاتی از `basket_b` که در قسمت `select` خواسته شده، در کنار اطلاعاتی از `basket_a` که در قسمت `Select` خواسته شده نمایش داده می‌شود. حال وقتی از `inner join`استفاده می‌کنیم، در واقع اشتراک دو جدول بر روی ستون میوه‌ها را نمایش می‌دهیم.اگر از `left join` استفاده کنیم، کل داده های `basket_a` نمایش داده می‌شود و قسمت‌هایی که در ستون `fruit_b` جدول دیگر، معادل دارند، در ستون‌های اضافی جدول مجازی ما قرار می‌گیرند و سایر جاها خالی می‌مانند. `Right_join` طبیعتا بر عکس است.

قسمت ب) براساس چیزی که از توضیحات استاد به خاطر دارم، `index` آخر جدول قرار است که کار جستجو را برای دیتا بیس را راحت کند و پرفرمنس آن را بالا برد. ساده‌ترین برداشت هم این است که برای این منظور از ستون‌هایی که تنها ارزش‌های منحصر به فرد در جدول می‌پذیرند برای این کار استفاده می‌کند.

قست ج)به طور کلی `subquery` حالتی است که از یک کوئری در کوئری دیگر استفاده می‌شود. دوباره توضیح یکی از مثال‌های لینک تدریس می‌تواند کمک کند.

```
Select
```

```
Film_id,
Title
From
Film
Where
film_id in(
select
inventory.film_id
from
rental
inner join inventory on inventory.inventory_id=rental.inventory_id
where
return_date between “2005-05-29” and “2005-05-30”
```

اول جدول دورنی تشکیل می‌شود، بین جداول rental و inventory یک join زده می‌شود و فیلم‌هایی که تاریخ آن در بازه خاصی قرار می‌گیرد، فیلتر می‌شوند، حال به ازای film_id هایی که در این جدول قرار دارند، ستون‌های film_id و title از جدول film استخراج می‌شوند.

قسمت د) در حالت کلی و ساده نرمال کردن به معنای تنظیم ساختار داده‌ها به گونه‌ای است که از تکرار غیرضروری اطلاعات جلوگیری شود. مثال قابل لمس آن همان چیزی است که سر کلاس مطرح شد. مثلا اگر دو جدول رابطه چند به چند داشته باشند، و بخواهیم آن‌ها را در یک جدول نمایش دهیم، ناچاریم که داده‌های تکراری زیادی نگه داریم، یکی از مسائلی که این رویکرد ایجاد می‌کند ناپایدار کردن دیتا بیس، سخت شدن به روزرسانی و اشغال کردن فضای اضافی است.
در طرف دیگر دینرمال کردن، به معنای آن است که عمدا برای بالا بردن کارایی در مواردی که مثلا یک دسته خاص از محاسبات نیاز است داده‌های تکراری را نگه داریم.
قسمت ه) یک جدول 1nf است اگر هر خانه آن فقط یک مقدار بگیرد.
یک جدول 2nf است اگر 1nf باشد و اصطلاحا هبستگی جزئی بین ستون‌های جدول و primary key وجود نداشته باشد. دقیقا مشابه همان مثالی که در کلاس مطرح شد. (مشکلی که وقتی تمام اطلاعات دو جدول با رابطه چند به چند را می خواستیم در یک جدول نگه داریم بروز می‌کرد)
یک جدول 3nf است، اگر 2nf باشد و هیچ رابطه تراگذاری نیز در آن وجود نداشته باشد. اگر مثال‌های کلاس را در نظر بگیریم، جدولی که یک رابطه یک به چند درشان خلاصه می‌شد، این ویژگی را نداشتند.
یک جدول bcnf است، اگر همه ستون‌های آن معرف رابطه باشند، مثلا جدولی میانی که برای رابطه چند به چند تعریف شده بود، اگر هیچ ستون دیگری نداشته باشد، یک جدول bcnf است.
قسمت و) دستور drop کل جدول با همه روابطی که برای آن تعریف شده و محتویاتش را پاک می‌کند، در حالی که truncate تنها محتویات جدول را پاک می‌کند.