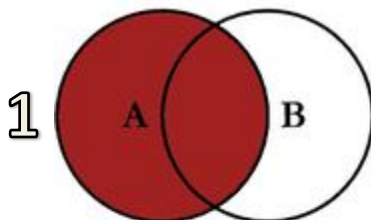
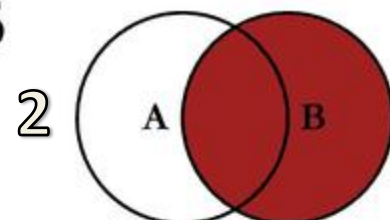


• فرق join, inner join, left join, right join

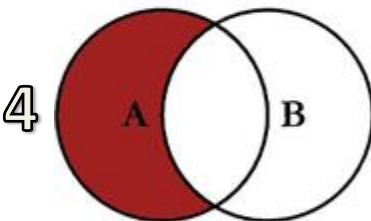
## SQL JOINS



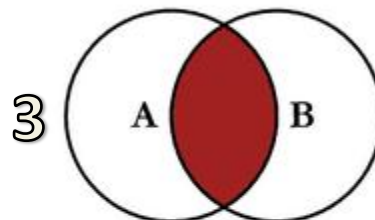
```
SELECT <select_list>  
FROM TableA A  
LEFT JOIN TableB B  
ON A.Key = B.Key
```



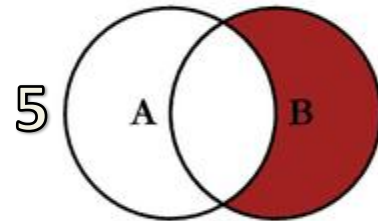
```
SELECT <select_list>  
FROM TableA A  
RIGHT JOIN TableB B  
ON A.Key = B.Key
```



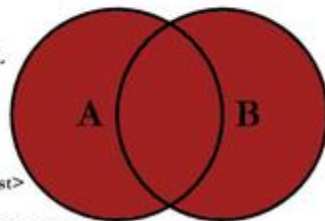
```
SELECT <select_list>  
FROM TableA A  
LEFT JOIN TableB B  
ON A.Key = B.Key  
WHERE B.Key IS NULL
```



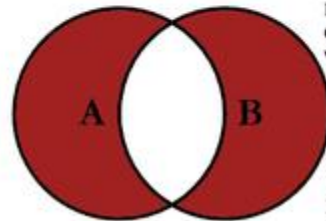
```
SELECT <select_list>  
FROM TableA A  
INNER JOIN TableB B  
ON A.Key = B.Key
```



```
SELECT <select_list>  
FROM TableA A  
RIGHT JOIN TableB B  
ON A.Key = B.Key  
WHERE A.Key IS NULL
```



```
SELECT <select_list>  
FROM TableA A  
FULL OUTER JOIN TableB B  
ON A.Key = B.Key
```



```
SELECT <select_list>  
FROM TableA A  
FULL OUTER JOIN TableB B  
ON A.Key = B.Key  
WHERE A.Key IS NULL  
OR B.Key IS NULL
```

6

7

### Join یا Inner join :

سطر هایی را برمیگرداند که در هر دو جدول تطابق دارد. یعنی ردیفی از جدول اول اگر در جدول دوم مقدار متناظر نداشته باشد به عنوان خروجی نمایش داده نمی شود. (شکل شماره سه)

### Left join :

تمام سطر هایی که در جدول چپ وجود دارد را برگردانده سپس سطر هایی که در جدول راست تطابق دارد را برمیگرداند و اگر سطر متناظر در جدول راست نبود Null برمیگرداند. (شکل شماره یک)

### Right join :

تمام سطر هایی که در جدول راست وجود دارد را برگردانده سپس سطر هایی که در جدول چپ تطابق دارد را برمیگرداند و اگر سطر متناظر در جدول چپ نبود Null برمیگرداند. (شکل شماره دو)

### full (outer) join :

تمام سطر های جدول راست و چپ را برگردانده و اگر متناظر ردیف های جدول راست در جدول چپ نبود و برعکس؛ Null برمیگرداند. (شکل شماره شش)

### • Index داخل دیتابیس چه کاربردی دارد؟

ایندکس باعث سریعتر اجرا شدن کوئری ها می شود. برای ساخت ایندکس می توان از دستور "create index" استفاده کرد. با ساخت ایندکس یک کپی از داده ها در جای دیگر ساخته می شود که مانند یک فهرست عمل کرده و به داده ها در داخل جدول اصلی اشاره دارند. همچنین هر ایندکس فضای ذخیره سازی مخصوص خودش را دارد. با اجرای update, delete, insert دیتابیس ایندکس ها رو مرتب نگه می دارد.

### • Subquery چیست؟

Subquery در واقع همان query است با این تفاوت که در داخل یک query دیگر نوشته شده است. برای مثال می توان داخل SELECT در یک query یک query دیگر نوشت و به آن Subquery میگویند.

## • Normalization و denormalization را در پایگاه داده توضیح دهید

**Normalization** به فرآیند حذف افزونگی و داده های زائد گفته میشود. هر گونه تکرار داده را حذف کرده و در نهایت هر entity دارای یک کلید اصلی می باشد که همه ی ستون ها در یک جدول تنها به این کلید تکیه می کنند. برای مثال اگر ستون در یک جدول با داده های تکراری وجود داشته باشد، از آن جدول حذف و به جدول دیگری منتقل میشود و با کلید اصلی به جدول اولیه وصل میگردد.

**Denormalization** به فرآیند اضافه کردن افزونگی که هدف از آن افزایش کارایی دیتابیس می باشد. Denormalization بعد از آن که داده ها نرمال شد انجام می گردد. این فرآیند نباید با داده هایی که نرمال نشده اند اشتباه گرفته شود. در واقع این فرآیند انجام می شود تا query زدن راحت تر و داده ها سریع تر خوانده شوند

## • شرح مختصری در مورد NF1, 2NF, 3NF, BCNF بدهید.

**NF1** : موجودیت و جدولی در فرم اول نرمال است که تمامی داده های آن بدون تکرار باشند. برای مثال اگر شخصی دارای دو شماره تلفن است نباید هر دو شماره در یک سلول وارد شود و باید در دو ردیف جدا تعریف گردد.

**NF2** : موجودیت و جدولی در فرم دوم نرمال است که اولاً شامل فرم اول نرمال باشد. دوماً تمامی ستون ها وابستگی کامل به کلید اصلی داشته باشند. برای مثال اگر کلید اصلیمان در یک جدول مرکب باشد باید با هر دو کلید وابستگی داشته باشد و نه با یکی از آن ها.

**NF3** : موجودیت و جدولی در فرم سوم نرمال است که اولاً شامل فرم دوم نرمال باشد. دوماً تمامی ستون ها باید فقط با کلید اصلی وابستگی داشته باشند. برای مثال اگر دو ستون غیر کلید اصلیمان به هم وابستگی داشته باشند یکی از آن ستون ها به جدول دیگری منتقل شده و با کلید اصلی به جدول اولیه مرتبط میگردد.

**BCNF** : موجودیت و جدولی در BCNF است که اولاً شامل فرم سوم نرمال باشد. در فرم دوم و سوم نرمال بحث بر سر وابستگی ستون های غیر کلیدی به کلید اصلی بود. در این فرم باید تمام ستون های غیر کلیدی آن کامل وابسته به یک کلید باشند و نه چیز دیگر . نکته این است که بحث بر سر وابستگی تابعی با یک کلید است نه فقط کلید اصلی. مفهوم فوق در خصوص موجودیت هائی که دارای چندین کلید هستند (Alternate Key) مطرح می شود . برای مثال اگر دو ستون به طور کامل به کلید اصل وابسته نباشند باید جدول تغییر کند و به نحوی تعریف شود که این وابستگی ایجاد شود. مثلاً جدولی با ستون های آیدی دانشجو، درس و استاد؛ باید به جدول آیدی استاد، نام استاد و درس تغییر کند و در جدولی دیگر با آیدی استاد به دانشجو متصل شود

- **DROP و TRUNCATE عملکردی مشخص دارند. تفاوت های دو دستور را بیان کنید.**

دستور DROP جهت حذف کامل جدول و محتویاتش استفاده میشود ولی دستور TRUNCATE جهت حذف تمام سطرهای جدول می باشد و خود جدول باقی می ماند.

در دستور DROP جدول حذف شده و فضای حافظه آزاد میشود ولی در دستور TRUNCATE خود جدول حذف نشده و فضایی آزاد نمیشود.

با اعمال دستور DROP جدولی قابل نمایش نیست ولی با اعمال دستور TRUNCATE خود جدول همچنان قابل نمایش است.

با اعمال دستور DROP تمام محدودیت ها همچون NULL پذیری، یکتایی، تایپ و اندازه داده ها و مقدار پیشفرض حذف میشود ولی با اعمال دستور TRUNCATE این محدودیت ها حذف نمی گردد.

دستور TRUNCATE سریع تر از دستور DROP می باشد.