

SQL (Structured Query Language) (زبان پرس و جو سافت یافته)

این زبان مبتنی بر جبر رابطه ای و حساب رابطه ای دامنه ای می باشد. این زبان در سال 1970 توسط شرکت IBM ارائه گردید. □

در سال 1986 استاندارد هائی توسط کمیته استاندارد آمریکا (ANSI) و کمیته استاندارد اروپا (ISO) بر آن اعمال شد. □

زبان SQL شامل موارد زیر می باشد.

DDL: زبان تعریف داده که SQL با استفاده از آن می تواند انواع دامنه و شماهای داده را تعریف کند.

DML: زبان دستکاری داده که SQL توسط این نوع دستورات، داده ها را عوض می کند، دستوراتی مانند *Delete, Update, Insert, Select* از

این نوع هستند.

این زبان شامل دستوراتی است که می تواند قوانین جامعیت را به پایگاه داده ارائه کند، همچنین شامل دستوراتی است که می توان شروع و فاصله تراکنش را مشخص کرد. □

دیر (view): دستوراتی که توسط آنها می توان رابطه (جداول) مجازی ایجاد کرد.

این زبان شامل دستوراتی جهت اعطای مجوز است که می توان به کاربران مقتلف امتیازات خاصی را اعطا کرد □

انواع داده در SQL (Data type):

Char(n): رشته ثابتی به طول n می باشد.

Varchar(n): رشته ای متغیر با طول حداکثر n می باشد.

int: تعریف عدد صحیح.

Small int: تعریف عدد صحیح کوچک (فضای آن به اندازه نیمی از فضای **int** می باشد).

numeric(p,d): داده ای عددی با طول میدان p، رقم و قسمت اعشاری d، رقم (عدد حقیقی).

Double Real: نوع داده اعشاری با دقت مضاعف.

Float(n): اعداد ممیز شناور با دقت n، رقم اعشار.

Date: بیانگر تاریخ می باشد و دارای ترتیب "ماه-روز-سال" می باشد.

Time: بیانگر زمان است و دارای فرمت "ثانیه:دقیقه:ساعت" می باشد.

Timestamp: ترکیبی از تاریخ و زمان با دقت میکرو ثانیه با نمایش 20 رقم دهدهی بدون علامت (yyyyymmddhhmmssnnnnn)

تابع **extract**:

این تابع جهت بدست آوردن فیلدهای خاص از داده های ذکر شده می باشد؛ به عنوان مثال اگر **d1** نام یک داده از نوع **Date** باشد

Extract(year from d1)

آنگاه دستور زیر تنها سال را از این نوع داده بر می گرداند.

□ به جای **year** می توان از **month**، **day** استفاده کرد.

تعریف داده بگرد.

`creat type Data-name as Reads type(s)`

در حالت کلی به این شکل می باشد

`create type color as ("Read","grean","blue")`

مثال.

در دستور صفحه قبل نوع داده *color* تعریف می شود و برای نوع این متغیر از داده های آماده *blue, Green, Red* استفاده می گردد با این تعریف مقدار داده *color* می تواند یکی از این سه مورد گفته شده باشد.

□ در *color f1, f1* از نوع *color* تعریف می شود و می تواند یکی از مقادیر *blue, Green, Red* را اختیار کند.

مثال. `creat type i as numeric(5,2)`

نوع داده ای بنام *i* تعریف می گردد که متغیر هاتی از نوع این داده می توانند اعدادی با 5 رقم صحیح و 2 رقم اعشار باشند.

حذف نوع داده تعریف شده.

`Drop type data name`

صورت کلی به این شکل می باشد.

مثال. `Drop type color` داده ای با نام *color* حذف می شود.

□ محدوده نوع دستوراتی که تعریف می کنیم تنها در میان دستورات می باشد

□ دستور معادل `Creat type` دستور `Domain–value Domain–name Creat Domain` می باشد.

مثال. `Creat Domain i (1,...,100)` نوع داده ای بنام *i* تعریف کرده ایم که تنها مقادیر بین 1 تا 100 را می پذیرد.

□ دستور معادل `Drop type` دستور `Drop Domain domain–name` می باشد. مثلاً با `Drop Domain i` نوع داده *i*

حذف می شود.

تعریف رابطه (جدول)

`Creat table table–name(A1d1...An dn)`

صورت کلی به این شکل می باشد.

A_1 تا A_n : نام صفات خاصه یا فیلدها می باشد

d_1 تا d_n : به ترتیب دامنه صفات خاصه A_1 تا A_n می باشد.

مثال. دستور `Creat table T1(ssn int, name char(30))` جدول مقابل را ایجاد می کند.

ssn	Name
:	:

T_1

`Creat table table–name (A1d1...An dn)`

[constraint 1]

:

[constraint m]

□ شکل توسعه یافته دستور قبل به صورت

می باشد، که موارد اضافه شده

محدودیت ها یا قید های اعمال شده بر روی رابطه می باشند.

تعریف کلید اصلی در رابطه: برای تعریف کلید اصلی دو روش وجود دارد

1. تعریف کلید اصلی بعد از تعریف صفت خاصه با استفاده از واژه *primary key* (مثال 1 صفحه بعد)

از این روش زمانی استفاده می شود که کلید اصلی ساده باشد و ترکیبی نباشد (اتمیک باشد)

2. تعریف کلید اصلی در قسمت محدودیت ها: از این روش زمانی استفاده می شود که کلید اصلی ترکیبی باشد (ترکیبی از چند صفت باشد).

البته کلید اصلی ساده را نیز می توان از این طریق تعریف کرد. (مثال 2 صفحه بعد)

```
Creat table t1(ssn int primery key,name char(30))
```

مثال 1.

```
Creat table t2(lname char[20],fname char[20],degree char[20])  
primery key(lname, fname)
```

مثال 2.

دستور بالا صفت ترکیبی *fname,lname* را به عنوان کلید اصلی می‌گیرد، که این کار بعد از تعریف فیلدها (صفات خاصه) و در قسمت ممبرودیت‌ها صورت می‌گیرد.

قید Not Null: اگر این قید (ممبرودیت) جلوی یک فیلد ظاهر شود، بدین معناست که مقدار این فیلد نمی‌تواند تهی باشد.

مثال.

```
creat table t3(id int not null , name char(30))
```

قید (p) chek: این قید در قسمت ممبرودیت‌ها ذکر می‌شود و بدین معناست که شرط *p* باید در جدول ذکر شده بر آورده شود.

```
Creat table mark (sn int primary key , grade numerice (4,2))  
check ((grade >= 00.00) and (grade < +20.00))
```

مثال.

قید $Unique(A_1 \dots A_j)$: این قید همراه صفت خاصه و یا در ممبرودیت‌ها ذکر می‌شود، و به این معناست که مقادیر صفات خاصه A_i تا A_j باید یکتا باشند. از این دستور در تعریف کلید کانزید استفاده می‌گردد.

مثال.

```
Creat table t(f1 char[20] , f2 char[30] , f3 int ,unique(f1,f2))
```

در دستور بالا مشخص می‌گردد که ترکیب مقادیر f_1 و f_2 نباید تکراری باشند.

پیاده سازی جامعیت ارباعی: برای پیاده سازی این جامعیت در قسمت ممبرودیت‌ها از دستور زیر استفاده می‌کنیم

```
foreign key (Ai) References Reference - Table  
[on Delete casecade]  
[on update casecade]
```

□ **Reference-table** جدول مربع است که در آن کلید خارجی به عنوان کلید اصلی می‌باشد

□ اگر مقدار کلید خارجی در جدول مربع حذف شود، و از **[on Delete casecade]** استفاده نشود **DBMS** جلوی حذف را می‌گیرد

ولی اگر از **[on Delete casecade]** استفاده شود عمل حذف در جدول مراجع کننده به صورت آبخاری انتشار می‌یابد.

□ **[on update casecade]** به این معناست که تغییرات بر روی کلید خارجی در جدول مربع به جدول رهیج کننده به صورت آبخاری

سرایت می‌کند ولی اگر این عبارت ذکر نشود **DBMS** جلوی تغییرات را می‌گیرد، البته می‌توان به جای این دو عبارت از عبارات

Set Default ,Set Null استفاده کرد، با استفاده از این دستورات با حذف و یا تغییر در کلید خارجی در جدول مربع، به جای آن کلید در

جدول مرتبط مقدار **NULL** یا مقدار پیش فرض قرار داده می‌شود.

```
creat table stud(sn int , name varchar(30) , city varchar(40) , ave numeric(4,2) , clg int)  
primery key(sn) , foreign key(clgn) , reference clg  
on Delet casecade  
on update casecade
```

مثال .

در دستور بالا رابطه **stud** ایبار شده و در قسمت صفات خاصه، صفات یا فیلدها همراه با دامنه شان آمده اند، و در قسمت ممبرودیت‌ها، **sn** از

نوع کلید اصلی بیان شده است، **clgn** به عنوان کلید خارجی تعریف شده که رابطه مربع برای **clgn** (رابطه ای که **clgn** در آن کلید اصلی است)

، **clg** بیان شده است و در آخر با بیان عبارات **on delete casecade , on update casecade** امکان حذف و بروز رسانی داده‌ها

در تمام جداول مرتبط با جدول مربع فراهم شده است

حذف جدول: دستور `Drop table table-name` جدول را به طور کامل حذف می کند و حتی شمای آن نیز باقی نماند.

مثال. `Drop table stud`.

تغییر شمای جدول:

□ دستور `Alter table table-name add Aidi`، فیلد A_i با دامنه d_i را به رابطه ای که نام آن در `table-name` ذکر شده اضافه می کند

□ دستور `Alter table table-name delete Ai`، فیلد A_i را از رابطه `table-name` حذف می کند

□ دستور `Alter table table-name modify Aidi`، فیلد A_i را از رابطه `table-name` به دامنه جدید d_i تغییر می دهد.

```
creat table t(ssn int , name char(30))
Alter table t add city char(50)
Alter table t delete name
```

مثال.

t

ssn	name
:	:

در خط اول دستور جدولی به نام *t* با دو فیلد `ssn, name` تشکیل می گردد

ssn	name	city
:	:	:

در خط دوم نام شهر (`city`) به فیلدهای جدول *t* اضافه می گردد.

ssn	city
:	:

در خط سوم فیلد `name` از شمای جدول حذف می گردد.

دستورات واقعی محیط SQL:

```
select part1 from part2
[where part3]
part4
```

می باشد.

دستور `select`: این دستور دارای سه قسمت است به این شکل

این دستور ترکیبی از عملگرهای تصویر (`part1`)، ضرب دکارتی (`part2`) و گزینش (`part3`) می باشد.

□ تمام فیلدهای جدول ذکر شده در `part2, part3, part4` را بر آورده می کند.

مثال. `select * from stud`.

sn	name	city	ave	clg
:	:	:	:	:

دستور بالا تمام تاپل های جدول *stud* با ذکر نام ستون هایش می باشد به این شکل

□ `part1` میتواند نام فیلد خاصی باشد، مثلا با دستور `select sn,ave from stud` تمام تاپل های ستون های `sn` و `ave` نمایش

داده می شود.

□ دستور `select` به طور عادی مقادیر تکراری در یک ستون یا فیلد را نمایش می دهد، برای جلوگیری از وارد شدن مقادیر تکراری از دستور `Distinct` قبل از نام فیلد استفاده می نمایم

مثال 1. دستور `select pname from sec` تمام مقادیر ستون `pname` را بدون توجه به تکراری بودن ارائه می دهد.

مثال 2. دستور `select Distinct pname from sec` نام های تکراری را تنها یک بار نمایش می دهد.

□ در `part1` میتوانیم برای یک فیلد نام مستعار داشته باشیم به این شکل

نام مستعار	<code>as</code>	نام فیلد(صفت خاصه)
------------	-----------------	--------------------

مثال . `select s#, city as shahr from stud`

S#	shahr
:	:

□ در `part1` می توان توابع مناسباتی قرار داد

مثال . `select sn, ave*1.2 from stud`، مقادیر `ave` در `1.2` ضرب می شوند و در همان فیلد قرار می گیرند