

انتخاب گزینه new و انتخاب فیلد کلید اصلی از هر دو جدول اصلی و فرعی و انتخاب تنظیمات مناسب از گزینه های پایین پنجره

روش دوم برای ایجاد ارتباط:

عملگر های join

cross join : توسط این نوع الحاق لیستی از تمام رکورد های جدول فرعی تولید می شود به طوریکه هر رکورد جدول اصلی با رکورد جدول فرعی نظیر به نظیر ارتباط دارد. (ارتباط یک به یک)

نام جدول ۲ cross join نام جدول ۱ from فیلد های جدول اصلی select

inner join : توسط این عملگر رکورد های از هر دو جدول طوری واکنشی می شود که مقادیر غیر null مرتبط را نمایش می دهد. (نوعی ارتباط یک به چند)

شرط on نام جدول ۲ inner join نام جدول ۱ from فیلد های جدول اصلی select

مثال: (فیلد شماره دانش آموزی به عنوان کلید اصلی در نظر گرفته شده است)

```
select student.std_name, student.std_family from
student innerjoin register on
student.std_num = register . std_num
```

: outer join

که شامل left, right, full می شود:

در این نوع الحاق همه رکورد های مرتبط با جدول اصلی که در جدول فرعی وجود دارد حتی مقادیر null واکنشی می شود.

left outer join : تمام رکورد های جدول اصلی را برمی گرداند و رکورد های مرتبط با جدول اصلی در جدول فرعی نمایش داده می شود و فیلدهایی که در جدول اصلی وجود دارد و در جدول فرعی وجود ندارد، null می شود.

right outer join : تمام رکورد های جدول فرعی را برمی گرداند و رکورد های مرتبط با جدول اصلی در جدول فرعی نمایش داده می شود و فیلد های که در جدول اصلی وجود ندارد ولی در جدول فرعی وجود دارد، null می شود.

full outer join : همه رکورد ها را برمی گرداند. در صورتی که یک رکورد از دو جدول نظیری نداشته باشد، null می شود.

جدول ۱ from نام فیلد ها select

شرط on جدول ۲ [left|right|full] outer join

8-دیدگاه ها

دید یا نما مجموعه ای از رکورد های بانک است که جهت دسترسی کاربران بانک دسته بندی می شود. از مهمترین مزایای دید امنیت داده ها و عدم دسترسی کاربران غیر مجاز به داده ها و تسریع دسترسی به داده ها است. view یک جدول مجازی است.

روش اول:

بر روی شی view در بانک اطلاعاتی راست کلیک کرده و گزینه new view را انتخاب می کنید.

روش دوم:

create view نام دید

select دستورات

نکته ۱: تمامی دستورات DDL (alter-drop-create) که روی جدول ها انجام می شود برای دید های نیز قابل انجام است.

نکته ۲: استفاده از دستور `order by` و `compute by` در `select` تشکیل دهنده `view` غیر مجاز است مگر اینکه کلمه `top` را به کار ببریم.

نکته ۳: استفاده از عبارت `select into` در دیدها غیر مجاز است.

مثال:

ایجاد یک دید از جدول دانش آموز فقط ۵ رکورد مرتبط شده بر اساس شماره که به ۳ ختم نشوند.

```
create view v_std as
select top 5 std_name, std_family , std_num
from student innerjoin register on
student.std_num = register.std_num
where std_num not like '%3'
```

9- تراکنش ها

تراکنش عبارت است از مجموعه ای از دستورات SQL که همه با هم اجرا می شوند یا لغو میشوند می تواند شامل `if` ، `for` ، `select` ... باشد.

ساختار کلی:

نام `begin transaction`

دستورات مورد نظر

`commit transaction`

مثال:

۲نمره به نمره دانش آموزان درس ریاضی در قالب `t.s` اضافه کنید.

`begin transaction t`

`update student set grade = grade+2`

```
where std_num in
(select std_num from register where lessen_name='math' )
commit transaction
```

در مقابل commit دستور rollback transaction رو داریم که همه دستورات مربوط به t.s را لغو می کند و بانک را به حالت قبل از اجرای آن برمی گرداند.

علاوه براین دو دستور (rollback) و (commit) دستور save transaction را نیز داریم که یک نقطه ذخیره در داخل t.s تعیین می کند. بعد از این کلمه نام یک قسمت از برنامه را می آوریم. بعد از آوردن کلمه save transaction معمولاً با دستور rollback به نقطه ذخیره برمی گردیم. (به اولین نقطه)

```
begin transaction t
...
save transaction
...
save transaction
...
rollback transaction
commim transaction
```

بعد از رسیدن به rollback به اولین نقطه ذخیره (save transaction1) برمی گردد.

تعریف متغیر در sql transaction ها

متغیر ها توسط دستور declare تعریف میشوند و توسط دستور set یا select مقدار دهی میشوند. به این صورت:

نوع داده نام متغیر @ declare

نکته: متغیر های محلی که با declare تعریف میشوند باید با علامت @ شروع شوند.

مثال (مقدار دهی متغیر با set):

پرس وجویی بنویسید که در آن نام تمام کسانی که شامل حرف a می شود برگردانده شود.

```
declare @name varchar
set @name = '%a%'
select std_name from student
where std_name like @name
```

مقدار دهی با **select**: این دستور حتما باید یک مقدار واحد را برگرداند.

```
declare @ave int
select @avg = ave(age) from student
print @avg
```

نکته: بادستور print میتوان محتویات یک متغیر را نمایش داد.

ساختارهای شرطی (**if, else**):

شکل کلی if ها در sql به صورت زیر میباشد.

if شرط یا همان دستورات

begin

دستورات

end

else

begin

دستورات

end

مثال: دستوری بنویسید که در صورت بزرگ بودن مجموع نمرات دانش آموز احمدی از ۱۰۰ پیغام مناسب نمایش دهد و

اگر نبود رکورد های مربوط به max و min نمرات نمایش داده شوند.