به نام خدا

دستور کار آزمایشگاه پایگاه داده

جلسه سوم

دوره‌ی آموزه‌های قبلی و نوشتن انواع پرس‌وجوهای پیچیده‌تر

استاد درس

دکتر شهریاری

تابستان ۹۹

مقدمه

در این جلسه ما ابتدا مروری خواهیم داشت بر دستورات پایه‌ای در SQL که از قبل آموزش دیده‌اید و با تنوع در پرس‌وجوها و جزییات بیشتر آن ها اشنا می‌شویم . سعی میکنیم تمام شروط و دستوراتی که در SQL کاربردی است و مهم هستند را به تفکیک و با توضیح یک مثال آموزش دهیم.

دوره‌ی مباحث پایگاه‌داده‌ی موردنیاز

Joins

دستور Join در SQL برای اتصال دو جدول استفاده می‌شود یعنی از هر جدول یک ستون انتخاب می‌کنیم تا به هم الحاق کنیم و این دو ستون شامل اعدادی هستند که درمیان دوجدول مشترک هستند.

Union clause

عملگر UNION برای ترکیب نتایج دو یا چند دستور SELECT استفاده می شود.

Sub Queries

منظور از Subquery یک query است که درون یک عبارت مانندUPDATE ، INSERT، SELECTیا DELETE نوشته می‌شود .

Group by

با استفاده از دستور group by می توان دسته بندی یا گروه بندی ستون ها را انجام داد.

تمرین پرس‌وجوهای پیچیده‌تر برای شروع

SELECT DISTINCT

چنانچه در ستون های مورد جستجو ، موارد تکراری وجود داشته باشد در نتیجه خروجی نمایش داده خواهند شد . برای جلوگیری از چنین موردی و عدم نمایش موارد تکراری پس از دستور Select عبارت DISTINCTنوشته می شود.

WHERE CONDITION

دستور Where برای اضافه کردن شرط یا شرط هایی جهت محدود کردن نتایج جستجو و یا استخراج نتایج دقیقتر برای داشتن خروجی که در ذهن ما وجود دارد استفاده می شود . این دستور باید پس از دستور Select و تعیین ستون ها از جدول مورد نظر به کار رود.

با استفاده از عملگرهای AND ، ORو پرانتز می توان چندین شرط را با هم ترکیب کرد .خروجی برنامه با شرط هایی که روی دستور داده

> نکته: همتای کلید واژه Distinct ، Allمی باشد که SQL Server را برای بازگرداندن همه سطرها آگاه می سازد خواه آن واحد باشد یا خیر All .پیش فرض دستور select است ، پس نیازی به نوشتن آن نیست.

شده است مطابقت داده خواهد شد .

AND و OR و =

عملگرهای And و Or و = برای ترکیب شرط ها در دستور Where در sql استفاده می شود.

گاهی اوقات خروجی که ما میخواهیم بایستی چند شرط مختلف داشته باشد . به طور مثال افرادی را میخواهیم که سن بالای ۲۳ سال و مدرک تحصیلی بالای لیسانس داشته باشند . در این حالت بایستی هر کدام از شرط ها را جداگانه تعریف کرده و سپس آنها را با هم ترکیب کنیم . برنامه هر کدام از شرط ها را بررسی میکند و خروجی را نمایش میدهد.

عملگر And برای اجرای دستور نیاز دارد تا تمام شرط های تعیین شده برای آن درست باشد.

عمگر Or فقط نیاز دارد که حداقل یکی از شرط ها درست باشد.

ترکیب قیود ANDو OR و = در شرطwhere :

```
1.      SELECT *
2.      FROM employees
3.      WHERE (city = 'Miami' AND first_name = 'Sarah')
4.      OR (employee_id <= 2000);
```

انواع JOIN

INNER JOIN

در این دستور تنها سطرهایی برمی‌گردند که حداقل یک ردیف در هردو جدول باشد که شرط پرس‌وجو را داشته باشد.

LEFT JOIN (LEFT OUTER JOIN)

در این دستور سطرهایی برمی‌گردند که داده‌هایشان در جدول چپ باشند، هرچند در جدول راست هیچ سطر منطبقی وجود نداشته باشد.

RIGHT JOIN (RIGHT OUTER JOIN)

در این دستور سطرهایی برمی‌گردند که داده‌هایی در جدول راست داشته باشند، هرچند در جدول چپ هیچ مورد منطبقی وجود نداشته باشد.

FULL JOIN

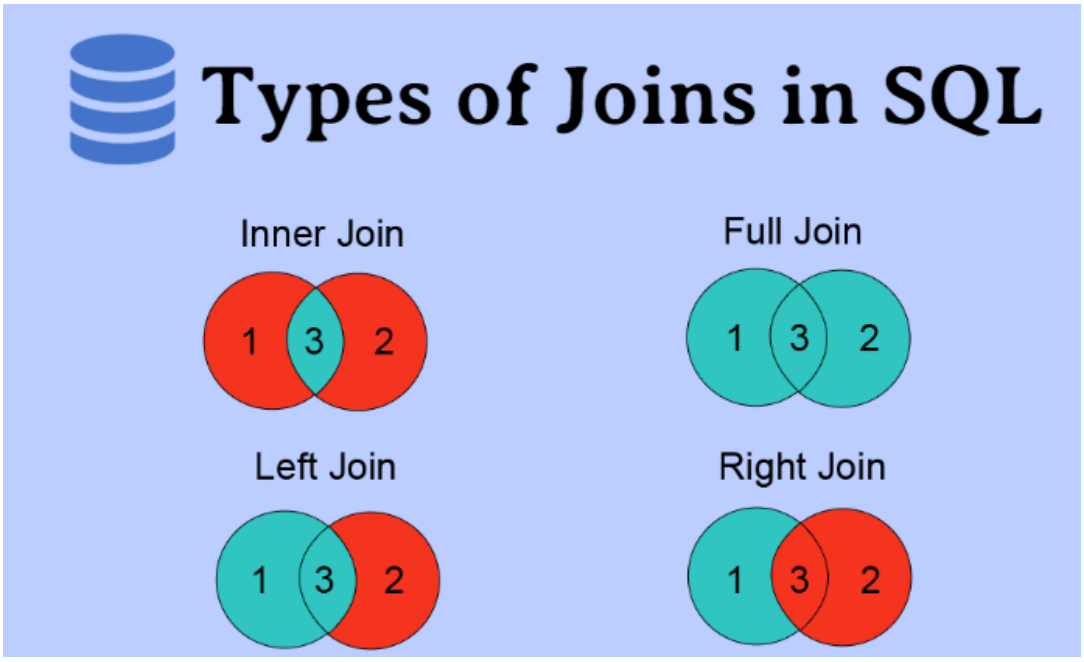
در این دستور همه سطرها تا زمانی‌که در یکی از جدول‌ها شرط پرس‌وجو برقرار باشد بازگشت داده می‌شوند.

مثال :

```
1.    Select  column1_name, column2_name, column3_name
2.    From    table1_name t
3.    inner/left/right/full join  table2_name b
4.    on  t.pk_column = b.pk_column
```

برای انتقال بهتر تفاوت بین JOIN ها توجه شمارا به تصویر زیر جلب می‌کنم.

بخش سبز رنگ، خروجی دستور از دو جدول مختلف نشان می‌دهد.



UNION

برای ترکیب و یا اجتماع گرفتن از دستورات SQL از این دستور استفاده می‌شود.

نکته: هر دستور SELECT در عملگر UNION باید تعداد ستون‌های یکسانی را برگردانند. همچنین ستون‌ها باید Data Type یکسانی داشته باشند. علاوه بر این ستون‌ها در هر دستور SELECT باید به یک صورت مرتب شده باشند.

```
1.      SELECT column1 [, column2 ]
2.      FROM table1 [, table2 ]
3.      [WHERE condition]
4.
5.      UNION
6.
7.      SELECT column1 [, column2 ]
8.      FROM table1 [, table2 ]
9.      [WHERE condition]
```

نکته UNION :تنها مقادیر غیر تکراری را انتخاب می کند. برای انتخاب مقادیر تکراری از UNION ALL استفاده کنید.

نکته: عملگر UNION تنها داده هایی را به صورت پیش فرض انتخاب می کند که از یکدیگر متمایز باشند. برای اینکه داده های تکراری را هم بیاوریم از کلید واژه ALL به همراه UNION استفاده کنید.

ORDER BY

اطلاعاتی که در دستور select در اسکیوال به عنوان خروجی نمایش داده میشود ی بی نظم و یا بهتر بگوییم بدون نظم مد نظر ما است. مقادیر خروجی در ستون های جدول بر اساس مقدار هیچ ستونی مرتب نمیشوند . با دستور دستور Order By میتوان اطلاعات جدول را بر اساس مقادیر یک یا چند ستون برحسب شاخص هایی مثل ترتیب حروف الفبا ، بزرگتر یا کوچکتر بودن اعداد و ...مرتب کرد.

```
1.      SELECT column_name(s)
2.      FROM table_name
3.      ORDER BY column_name(s) [ASC|DESC]
```

علامت [] در اطراف where بدین معناست که میتوانید where را بکار نبرید. اما اگر بکار بردید حتما باید قبل از order باشد .

ASCبه معنای صعودی بودن(a to z) و DESC به معنای نزولی بودن است.(z to a) پیش فرض ACS است.

نکته: همچنین این نیز امکان پذیرست که مرتب سازی را بر مبنای بیش از یک ستون انجام دهید

GROUP BY

با استفاده از دستور group by می توان دسته بندی یک ستون بر حسب مقادیر مشابه فیلدهای یک ستون دیگر.را انجام داد. در هنگام استفاده از برخی از توابع درون ساخته SQL که عمل محاسبه (مثل مجموع و میانگین) را بر روی داده ها انجام می دهند ، این مشکل وجود دارد که این توابع قادر به جدا کردن و متمایز کردن اطلاعات موجود در دو ستون نسبت به هم نیستند و نتایج محاسبات را به صورت کلی برای همه آنها در نظر می گیرند . در این مواقع از دستور Group By استفاده میکنیم.

نکته: گروه‌بندی را نباید روی ستون کلید اصلی جدول قرار داد چون تمام مقادیر یکتا هستند و آنگاه هر سطر را به یک گروه تبدیل می‌کند.

```
1.     SELECT column1, column2
2.     FROM table_name
3.     WHERE [ conditions ]
4.     GROUP BY column1, column2
5.     ORDER BY column1, column2
```

نکته: دستور Group By در sql وقتی استفاده میشود که ما در حال انتخاب چند ستون هستیم و حداقل یک عملگر محاسباتی در دستور select داریم . در این زمان ما باید تمام ستونهای دیگر را گروه کنیم..

نکته:ORDER BY دستوری برای مرتب‌سازی خروجی بر اساس یک یا چند ستون است.

HAVING

دستور Having برای افزودن شرط به توابع درون ساخته SQL استفاده می شود ، زیرا از دستور Where نمی توان برای کار با مقادیر خروجی توابع درون ساخته SQL استفاده کرد . به عبارت دیگر دستور Having در sql برای اعمال شرط به ستون ها اعمال می شود و همان کاری را می کند که Where در رکوردها انجام می دهد . دستور Having معمولا با دستور Group By می آید.

از دستور HAVING به شکل زیر استفاده می‌شود:

```
SELECT column_name, aggregate_function(column_name)
FROM table_name
WHERE column_name operator value
GROUP BY column_name
HAVING aggregate_function(column_name) operator value
```

مثال دیگری برای درک بهتر استفاده از HAVING همرا با تابع sum و اعمال شرط بیشتر از ۲۰۰:

```
SELECT
    customer_id,
    SUM (amount)
FROM
    payment
GROUP BY
    customer_id
HAVING
    SUM (amount) > 200;
```

WITH CLAUSE

With همان دستور CTE است و CTE همان جدول مجازی.

CTE = Common Table Expression

برای ساخت CTE از کلمه کلیدی WITH استفاده می‌کنیم و یک نام برای CTE می‌گذاریم و SELECT مدنظر را اجرا می‌کنیم.

```
WITH query_name (column_name1, ...) AS
     (SELECT ...)
```

نکته: عمر جدول مجازی CTE فقط محدود به یک دستور است. دستوری که بلافاصله بعد از CTE نوشته می‌شود.

مثال‌های دیگری از WITH

```
WITH cte_film AS (
    SELECT
        film_id,
        title,
        (CASE
            WHEN length < 30 THEN 'Short'
            WHEN length < 90 THEN 'Medium'
            ELSE 'Long'
        END) length
    FROM
        film
```

```
)
SELECT
    film_id,
    title,
    length
FROM
    cte_film
WHERE
    length = 'Long'
ORDER BY
    title;
```

برای ترکیب کردن عبارت WITH با یک جدول

```
WITH cte_rental AS (
    SELECT staff_id,
        COUNT(rental_id) rental_count
    FROM   rental
    GROUP  BY staff_id
)
SELECT s.staff_id,
    first_name,
    last_name,
    rental_count
FROM staff s
    INNER JOIN cte_rental USING (staff_id);
```

در این مثال ابتدا CTE یم نتایج شامل staff id و تعداد rental را برمی‌گرداند سپس جدول staff را بادستور cte با استفاده از ستون staff id ترکیب می‌کند.

SUBQUERY یا NESTED QUERY

وقتی یک ساختار SQL داخل یک ساختار SQL قرار می‌گیرد Subquery نامیده می‌شود.

> نکته : به SQL که داخل آن یک SQL دیگر قراردارد Parent statement هم می‌گویند.

```
    SELECT column_name [, column_name ]
    FROM   table1 [, table2 ]
    WHERE  column_name OPERATOR
       (SELECT column_name [, column_name ]
       FROM table1 [, table2 ]
       [WHERE])
```

نمونه یک پرس‌وجو در یک پرس‌وجوی دیگر:

```
    SELECT
        column_1
        , column_2
        , column_3
    FROM
        tbl_data
    WHERE
        column_1 IN -
- this can also be "NOT IN", "EXISTS, an operator like "=", "<", and others.
        (
        SELECT
            column_1
        FROM
            tbl_data
        WHERE
            [condition]
        )
    ORDER BY column_1
```

دستورات قابل استفاده در پرس‌وجو ها:

IN & NOT IN

```
SELECT *
FROM employees
WHERE last_name NOT IN/IN ('Anderson', 'Johnson', 'Smith');
```

IS NULL

```
SELECT *
FROM contacts
WHERE address IS NOT NULL;
```

ANY

```
SELECT count(aid),bid FROM pgbench_accounts WHERE
bid = ANY(SELECT bid FROM pgbench_branches WHERE bbalance > 0)
GROUP BY bid;
```

EXIST/NOT EXIST

```
SELECT *
FROM products
WHERE NOT EXISTS (SELECT 1
                FROM inventory
                WHERE products.product_id = inventory.product_id);
```

ALL

```
SELECT count(aid),bid FROM pgbench_accounts WHERE
bid <> ALL(SELECT bid FROM pgbench_branches WHERE bbalance > 0)
```

```
3.  GROUP BY bid;
```

LIKE

دستور LIKE برای پیدا کردن تشابه در متن و مقدار STRING در پرس‌وجو استفاده می‌شود و با دو کاراکتر % و ـ می‌آید.

“ ٪” نشان دهنده صفر یا یک یا چند کاراکتر است و “_” نشان دهنده تک کاراکتر است و می‌توانند باهم نیز استفاده شوند

```
1.  SELECT FROM table_name
2.  WHERE column LIKE 'XXXX%'
3.
4.  or
5.
6.  SELECT FROM table_name
7.  WHERE column LIKE '%XXXX%'
8.
9.  or
10.
11. SELECT FROM table_name
12. WHERE column LIKE 'XXXX_'
13.
14. or
15.
16. SELECT FROM table_name
17. WHERE column LIKE '_XXXX'
18.
19. or
20.
21. SELECT FROM table_name
22. WHERE column LIKE '_XXXX_'
```

نکته : X ها در اینجا می‌توانند هر مقدار رشته (STRING) باشد.

تمرین

۱-از ترکیب دستورات GOUP BY و HAVING دو پرس‌وجوی متفاوت بزنید و خروجی آن در جداول خودتان را بفرستید و آن را توضیح دهید.

۲-یک دستور مرتب سازی شده براساس دو ستون خاص با ترکیب union بزنید و خروجی آن را بفرستید.

۳-یک عبارت CTE تعریف کنید و با یک دستور پرس‌وجوی دیگر ترکیب کنید و خروجی آن را بفرستید.

۴-با دو دستور مختلف دو تا از join های مختلف را استفاده کنید که خروجی هایشان متفاوت باشد و توضیح دهید.

۵- دو دستور تو در تو بنویسید که ۳ دستور را باهم ترکیب کند و حتما در خروجی از تک تک دستورات any,<,>,is null, like, all, exist استفاده کند.

نکات مهم هنگام تمرین:

- کلیه مستند تکلیف‌ها (کد و گزارش و خروجی ها) باید در قالب یک فایل فشرده با نام **DBlab_#studentnumber_#assignmentnumber** (عدد آخر شماره سری تمرین است) در سامانه courses اپلود شوند.
- درصورت دستی نوشتن تکالیف تشریحی و ناخوانا بودن آن‌ها باعث کسر نمره می‌شود.
- درصورت بروز اشکال درمورد هر تمرین تا ۲۴ ساعت قبل از اتمام زمان ارسال، به آدرس **Maryam.taherig@gmail.com** ایمیل بزنید.
- تاخیر در ارسال تکالیف به ازای هرروز مشمول ۱۰٪ کسر نمره خواهد شد.
- درصورت کشف تقلب براساس مقررات آموزشی برخورد خواهد شد.
- حداکثر تعداد صفحات باید ۱۰ صفحه باشد.

موفق باشید