10.07 메모

Created	@2024년 10월 7일 오전 9:25
Class	9.30~10.10 데이터베이스 구축

지필 평가

관계형 데이터베이스
(Athena, PostgreSQL) 11문항 (객관식 7문항, 단답형 3문항, 서술형 1문항)
비관계형 데이터베이스
(MongoDB) 4?문항 (객관식 3문항, 서술형 1문항)
비관계형 데이터베이스

(redis) 1문항 (서술형 1문항)

총 15문항

```
select
proc_ymd, mbr_sex_cd_nm, count(userid) as user_cnt
from "learning_analytics"."e_milkt_study"
where 1=1
and "proc_ym" = '202409'
and "proc_ymd" is not null
and "mbr_sex_cd_nm" is not null
group by proc_ymd, mbr_sex_cd_nm
order by proc_ymd, mbr_sex_cd_nm
```

이렇게 하면 proc_ymd로 그룹화 이후, 그 안에서 또 mbr_sex_cd_nm으로 또 그룹화한 것이다.

```
select
proc_ymd, mbr_sex_cd_nm, count(userid) as user_cnt
from "learning_analytics"."e_milkt_study"
where 1=1
```

```
and "proc_ym" = '202409'
and "proc_ymd" is not null
and "mbr_sex_cd_nm" is not null
group by grouping sets (proc_ymd, mbr_sex_cd_nm)
order by proc_ymd, mbr_sex_cd_nm
```

grouping sets는 proc_ymd로 그룹화 한 결과를 일단 출력하고, 그 밑에 mbr_sex_cd_nm으로 그룹화 한 결과를 출력한다.

(두 컬럼으로 그룹화한 것이 아니라 하나로 그룹화한 결과 각각을 출력한다.)

```
select
proc_ymd, mbr_sex_cd_nm, count(userid) as user_cnt
from "learning_analytics"."e_milkt_study"
where 1=1
and "proc_ym" = '202409'
and "proc_ymd" is not null
and "mbr_sex_cd_nm" is not null
group by rollup (proc_ymd, mbr_sex_cd_nm)
order by proc_ymd, mbr_sex_cd_nm
```

rollup은 먼저 두 컬럼으로 그룹화 한 결과도 출력하고, 컬럼 값으로 구분했던 값들의 전체 합도 출력한다.

1	20240901	남	72320
2	20240901	여	53159
3	20240901	정보없음	5369
4	20240901		130848

```
select
proc_ymd, mbr_sex_cd_nm, count(userid) as user_cnt
from "learning_analytics"."e_milkt_study"
where 1=1
and "proc_ym" = '202409'
and "proc_ymd" is not null
and "mbr_sex_cd_nm" is not null
group by cube (proc_ymd, mbr_sex_cd_nm)
order by proc_ymd, mbr_sex_cd_nm
```

cube는 두 컬럼으로 그룹화한 결과와, 각각 한 개의 컬럼으로만 그룹화한 결과들도 출력한다.

```
select
proc_ymd, userid, cjt050_timestamp,
rank() over(partition by proc_ymd, userid order by cjt050_tim
from learning_analytics.e_media
where 1=1
and proc_ymd = '20221129'
and userid = '000552bd-54cb-4ddd-a4a1-8549214a1e64'
order by rank
```

nulls first/last : null 값이 맨 위/뒤에 나온다.

테이블을 만들 때 유의할 점

● 테이블 명은 다른 테이블의 이름과 중복되지 않아야 한다.

● 한 테이블 내에서는 컬럼명이 중복되게 지정될 수 없다.

● 테이블 이름을 지정하고 각 컬럼들은 괄호 "()" 로 묶어 지정한다.

● 각 컬럼들은 콤마(,) 로 구분되고, 테이블 생성문의 끝은 항상 세미콜론(;) 으로 끝난다.

● 컬럼 뒤에 데이터 유형은 꼭 지정되어야 한다.

● 테이블명과 컬럼명은 반드시 문자로 시작해야 하고, 길이 제한이 있는 경우가 있다.

● A-Z, a-z, 0-9, _, $, # 문자만 허용된다.

DDL & DML

```
CREATE TABLE today_study (
    today char(8),
    summary varchar,
    note varchar
);

ALTER TABLE today_study ADD COLUMN idx int;
```

```
DROP TABLE today_study;

INSERT INTO today_study (today, summary, note, idx)
VALUES ('20240930', '오리엔테이션', '    ', 1);

INSERT INTO today_study (today, summary, note, idx)
VALUES
('20240930', '관련직무', '    ', 2),
('20240930', '환경구성', '    ', 3),
('20240930', '데이터의 위치', '    ', 2),
('20240930', '데이터의 종류', '    ', 3);

CREATE TABLE today_study_backup AS
SELECT *
FROM today_study
WHERE 0=1;

INSERT INTO today_study_backup
SELECT *
FROM today_study;

UPDATE today_study
SET idx = 4
WHERE summary = '데이터의 위치';

UPDATE today_study
SET idx = 5
WHERE summary = '데이터의 종류';

UPDATE today_study
SET note = NULL
WHERE note = '    ';

DELETE FROM today_study
WHERE summary = '데이터의 위치';

DELETE FROM today_study;
```

```
CREATE TABLE today_study_backup_2 AS
SELECT *
FROM today_study_backup
WHERE 1=1;

TRUNCATE TABLE today_study_backup;
```

제약조건 걸기, 트랜젝션 모드 manual로 하기

```
CREATE TABLE account (
    user_id serial PRIMARY KEY,
    user_name varchar(50) UNIQUE NOT NULL,
    PASSWORD varchar(50) NOT NULL,
    email varchar(500) UNIQUE NOT NULL,
    created_on timestamp NOT NULL,
    last_login timestamp
);

INSERT INTO account VALUES (1, '곽태경', '123123', 'xorud755@gm
INSERT INTO account VALUES (2, '곽태경', '123123', 'xorud755@gm
INSERT INTO account VALUES (2, '곽태', '12312', 'xo755@gmail.c

ALTER TABLE account ADD CONSTRAINT "unique_password" UNIQUE (
ALTER TABLE account DROP CONSTRAINT "unique_password";
ALTER TABLE account ALTER COLUMN "email" DROP NOT NULL;
ALTER TABLE account ALTER COLUMN "email" SET NOT NULL;
ALTER TABLE account DROP CONSTRAINT "account_pkey";

UPDATE account SET user_id = 2 WHERE user_name = '곽태';

ALTER TABLE account ADD CONSTRAINT "account_pkey" UNIQUE ("us
ALTER TABLE account ADD CONSTRAINT "check_password" check ("p
ALTER TABLE account DROP CONSTRAINT "check_password";
DROP TABLE account;

CREATE TABLE account (
```

```
    USER_ID SERIAL PRIMARY KEY,
    USER_NAME VARCHAR(50) UNIQUE NOT NULL,
    PASSWORD VARCHAR(50) NOT NULL,
    EMAIL VARCHAR(500) UNIQUE NOT NULL,
    CREATED_ON TIMESTAMP NOT NULL,
    LAST_LOGIN TIMESTAMP
);

CREATE TABLE role (
    ROLE_ID SERIAL PRIMARY KEY,
    ROLE_NAME VARCHAR(255) UNIQUE NOT NULL
);

CREATE TABLE account_role (
    USER_ID INTEGER NOT NULL,
    ROLE_ID INTEGER NOT NULL,
    GRANT_DATE TIMESTAMP WITHOUT TIME ZONE,
    PRIMARY KEY (USER_ID, ROLE_ID),
    CONSTRAINT ACCOUNT_ROLE_ROLE_ID_FKEY FOREIGN KEY (ROLE_ID
        REFERENCES ROLE (ROLE_ID) MATCH SIMPLE
        ON UPDATE NO ACTION ON DELETE NO ACTION,
    CONSTRAINT ACCOUNT_ROLE_USER_ID_FKEY FOREIGN KEY (USER_ID
        REFERENCES ACCOUNT (USER_ID) MATCH SIMPLE
        ON UPDATE NO ACTION ON DELETE NO ACTION
);

INSERT INTO account VALUES (1, 'rhkrxorud', '123456', 'aaa@bb
INSERT INTO ROLE VALUES (1, '학생');
INSERT INTO account_ROLE VALUES (1,1, current_timestamp);
INSERT INTO account_ROLE VALUES (2,1, current_timestamp); --

UPDATE "role" SET role_id = 2 where role_id = 1; -- fkey 라서
DELETE FROM ROLE WHERE role_id = 3;

BEGIN; -- 트랜젝션 manual로 바꿈
INSERT INTO ROLE VALUES (3, 'friend');
ROLLBACK;
INSERT INTO ROLE VALUES (3, 'friend');
```

```
SAVEPOINT "mysave";
INSERT INTO ROLE VALUES (4, 'qwerasdf');
rollback TO SAVEPOINT "mysave";
COMMIT;
```