Day69; 20221214

날짜	@2022년 12월 14일
유형	@2022년 12월 14일
태그	

GitHub - u8yes/AI

You can't perform that action at this time. You signed in with another tab or window. You signed out in another tab or window. Reload to refresh your session. Reload to refresh your session.

https://github.com/u8yes/ai

u8yes/**AI**

1 Contributor 0 Issues 0 Stars 0 Forks

https://s3-us-west-2.amazonaws.com/secure.notion-static.com/b4e348f2-17d4-45fb-8bc2-a96f731c860d/scott.sql

https://s3-us-west-2.amazonaws.com/secure.notion-static.com/c42297ae-1e61-4403-9685-c3e0edc8ea6e/%EB%A8%B8%EC%8B%A0%EB%9F%AC%EB%8B%9D_%ED%94%84%EB%A1%9C%EC%A0%9D%ED%8A%B8_%EC%A7%84%ED%96%89_%EC%A0%88%EC%B0%A8.pdf

https://s3-us-west-2.amazonaws.com/secure.notion-static.com/bcdad170-daf7-4924-bdb5-1dd3e7e1a440/%EC%A1%B0%EC%9D%B8.pdf

---

조인(join)

```
Rem Copyright (c) 1990 by Oracle Corporation
Rem NAME
REM    UTLSAMPL.SQL
Rem  FUNCTION
Rem  NOTES
Rem  MODIFIED
Rem gdudey     06/28/95 -  Modified for desktop seed database
Rem glumpkin   10/21/92 -  Renamed from SQLBLD.SQL
Rem blinden   07/27/92 -  Added primary and foreign keys to EMP and DEPT
Rem rlim     04/29/91 -       change char to varchar2
Rem mmoore     04/08/91 -       use unlimited tablespace priv
Rem pritto     04/04/91 -       change SYSDATE to 13-JUL-87
Rem   Mendels  12/07/90 - bug 30123;add to_date calls so language independent
Rem
rem
rem $Header: utlsampl.sql 7020100.1 94/09/23 22:14:24 cli Generic<base> $ sqlbld.sql
rem
SET TERMOUT OFF
SET ECHO OFF

rem CONGDON    Invoked in RDBMS at build time.   29-DEC-1988
rem OATES:     Created: 16-Feb-83

GRANT CONNECT,RESOURCE,UNLIMITED TABLESPACE TO SCOTT IDENTIFIED BY TIGER;
ALTER USER SCOTT DEFAULT TABLESPACE USERS;
ALTER USER SCOTT TEMPORARY TABLESPACE TEMP;
CONNECT SCOTT/TIGER
DROP TABLE DEPT;
CREATE TABLE DEPT
       (DEPTNO NUMBER(2) CONSTRAINT PK_DEPT PRIMARY KEY,
  DNAME VARCHAR2(14) ,
  LOC VARCHAR2(13) ) ;
DROP TABLE EMP;
CREATE TABLE EMP
       (EMPNO NUMBER(4) CONSTRAINT PK_EMP PRIMARY KEY,
  ENAME VARCHAR2(10),
  JOB VARCHAR2(9),
  MGR NUMBER(4),
  HIREDATE DATE,
  SAL NUMBER(7,2),
  COMM NUMBER(7,2),
  DEPTNO NUMBER(2) CONSTRAINT FK_DEPTNO REFERENCES DEPT);
```

```
INSERT INTO DEPT VALUES
  (10,'ACCOUNTING','NEW YORK');
INSERT INTO DEPT VALUES (20,'RESEARCH','DALLAS');
INSERT INTO DEPT VALUES
  (30,'SALES','CHICAGO');
INSERT INTO DEPT VALUES
  (40,'OPERATIONS','BOSTON');
INSERT INTO EMP VALUES
(7369,'SMITH','CLERK',7902,to_date('17-12-1980','dd-mm-yyyy'),800,NULL,20);
INSERT INTO EMP VALUES
(7499,'ALLEN','SALESMAN',7698,to_date('20-2-1981','dd-mm-yyyy'),1600,300,30);
INSERT INTO EMP VALUES
(7521,'WARD','SALESMAN',7698,to_date('22-2-1981','dd-mm-yyyy'),1250,500,30);
INSERT INTO EMP VALUES
(7566,'JONES','MANAGER',7839,to_date('2-4-1981','dd-mm-yyyy'),2975,NULL,20);
INSERT INTO EMP VALUES
(7654,'MARTIN','SALESMAN',7698,to_date('28-9-1981','dd-mm-yyyy'),1250,1400,30);
INSERT INTO EMP VALUES
(7698,'BLAKE','MANAGER',7839,to_date('1-5-1981','dd-mm-yyyy'),2850,NULL,30);
INSERT INTO EMP VALUES
(7782,'CLARK','MANAGER',7839,to_date('9-6-1981','dd-mm-yyyy'),2450,NULL,10);
INSERT INTO EMP VALUES
(7788,'SCOTT','ANALYST',7566,to_date('13-7-1987','dd-mm-yyyy'),3000,NULL,20);
INSERT INTO EMP VALUES
(7839,'KING','PRESIDENT',NULL,to_date('17-11-1981','dd-mm-yyyy'),5000,NULL,10);
INSERT INTO EMP VALUES
(7844,'TURNER','SALESMAN',7698,to_date('8-9-1981','dd-mm-yyyy'),1500,0,30);
INSERT INTO EMP VALUES
(7876,'ADAMS','CLERK',7788,to_date('13-7-1987','dd-mm-yyyy'),1100,NULL,20);
INSERT INTO EMP VALUES
(7900,'JAMES','CLERK',7698,to_date('3-12-1981','dd-mm-yyyy'),950,NULL,30);
INSERT INTO EMP VALUES
(7902,'FORD','ANALYST',7566,to_date('3-12-1981','dd-mm-yyyy'),3000,NULL,20);
INSERT INTO EMP VALUES
(7934,'MILLER','CLERK',7782,to_date('23-1-1982','dd-mm-yyyy'),1300,NULL,10);
DROP TABLE BONUS;
CREATE TABLE BONUS
  (
  ENAME VARCHAR2(10)  ,
  JOB VARCHAR2(9)  ,
  SAL NUMBER,
  COMM NUMBER
  ) ;
DROP TABLE SALGRADE;
CREATE TABLE SALGRADE
      ( GRADE NUMBER,
  LOSAL NUMBER,
  HISAL NUMBER );
INSERT INTO SALGRADE VALUES (1,700,1200);
INSERT INTO SALGRADE VALUES (2,1201,1400);
INSERT INTO SALGRADE VALUES (3,1401,2000);
INSERT INTO SALGRADE VALUES (4,2001,3000);
INSERT INTO SALGRADE VALUES (5,3001,9999);
COMMIT;

SET TERMOUT ON
SET ECHO ON
```

- 두 개 이상의 테이블에 나뉘어져 있는 데이터를 한 번의 sql문으로 원하는 결과를 얻을 수 있는 기능(이름이 scott인 사원의 부서명 출력).
- 종류
  1) cross join : 2개 이상의 테이블이 조인될 때 where절에 의해 공통되는 컬럼에 의한 결합이 발생하지 않는 경우를 의미.
  2) equi join : 조인 대상이 되는 두 테이블에서 공통적으로 존재하는 컬럼의 값이 일치되는 행을 연결하여 결과를 생성하는 방법.
  3) non-equi join : 동일 컬럼이 없이 다른 조건을 사용하여 join. 조인 조건에 특정 범위 내에 있는지를 조사하기 위해서 조건절에 조인 조건을 = 연산자 이외의 비교 연산자를 이용.
  4) self join : 하나의 테이블 내에서, 자기 자신과 조인을 통해 원하는 자료를 얻는 방법.
  5) outer join : 조인 조건에 만족하지 못해서 해당 결과를 출력시에 누락이 되는 문제점이 발생. 해당 레코드(row)를 출력하고 싶을 때 사용하는 join 방법.

Join 종류

종 류	설 명
Equi Join	동일 칼럼을 기준으로 조인합니다.
Non-Equi Join	동일 칼럼이 없이 다른 조건을 사용하여 조인합니다.
Outer Join	조인 조건에 만족하지 않는 행도 나타낸다.
Seif Join	한 테이블 내에서 조인합니다.

ANSI 조인

1. Ansi Cross join
2. Ansi Inner join
3. Ansi Outer join

1.6 데이터 분석 프로젝트가 실무에서는 어떻게 이루어질까?

1.6.2 유통

유통업 분야에서도 데이터는 중요하게 활용된다. 예를 들면, 배송업체에서 데이터 분석을 어떻게 적용할 수 있을까? 배송 앱을 통해 수집된 수만 개의 주문 내역들, 주문 건수, 재고량 등의 데이터를 축적하여 가장 효율적인 배송 방법을 설계할 수 있다. 혁신적인 기업들은 고객 주문 데이터와 재고 현황을 30분 단위로 전 직원에게 업데이트하여 유동적으로 변하는 고객의 주문량에 대해 더욱 신속히 대응한다. 이러한 정보는 고객에게도 앱을 통해 공유된다. 고객은 물건을 주문한 시점부터 회사에서 물건을 준비하는 시간, 배송 기사들의 배송 경로, 예상 도착 시각 등을 살펴볼 수 있어 고객의 만족도를 증진시킨다. 예를 들어, 쿠팡의 경우 배송 준비 단계부터 고객에게 상품이 전달되는 모든 과정을 데이터로 수집하여 '로켓배송'이라는 혁신적인 비즈니스 전략을 만들었다. 이는 고객 만족도 최고 96점이라는 높은 성과를 만들었고, 쿠팡의 2020년 매출은 2014년 매출과 비교하여 약 40배 성장하였다.

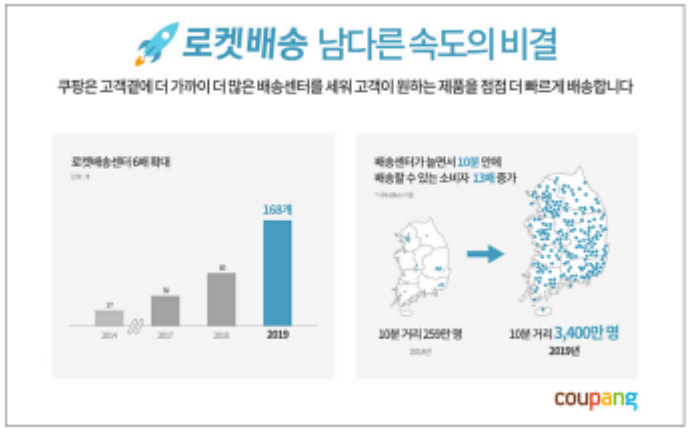

그림 1.4 데이터를 기반으로 한 쿠팡의 로켓배송

출처 https://bit.ly/3o1YWOR

1.6.3 공공

공공 분야에서도 정부와 각 행정 지자체별로 데이터는 활발하게 사용된다. 예를 들면, 서울시는 방대한 데이터를 바탕으로 심야버스, 일명 '올빼미버스' 노선을 지정한 바 있다. 심야 유동인구의 스마트카드 사용 내역을 기반으로 택시의 승하차 데이터를 얻고, 통화 이력 데이터를 활용하여 위치 정보를 취득하였다. 이러한 데이터를 통해 유동인구의 지역별 밀도를 구체적으로 분석이 가능하였고, 보다 과학적인 버스 노선도를 만들었다. 이렇게 공공 분야에서도 데이터를 활용해서 보다 효율적이고 정확한 행정 업무들을 진행하고 있는 추세다.

그림 1.5 데이터를 기반으로 심야버스 노선도, 배차 간격, 정류장 결정 과정

출처 https://bit.ly/3CGKcJl

머신러닝 프로젝트 진행 절차

머신러닝 프로젝트 진행 절차

1. 문제 정의 : 비즈니스 문제 정의
2. 데이터 수집 : 데이터 세트를 식별하고 수집
3. 데이터 준비 : 데이터로부터 통찰을 얻기 위해 탐색하고 시각화
   - 결측치 대치
   - 중복 레코드 제거
   - 값 정규화
   - 다른 유형의 정리나 매핑 수행
   - 완전한 특성 추출
   - 상관 특성 제거
4. 데이터 분리 : 데이터를 훈련 세트, 검증 세트, 테스트 세트로 분할
5. 모델 학습 : 훈련 데이터 세트를 사용해 기계 모델은 훈련
6. 후보 모델 평가 : 모델 정확도를 결정하기 위해 테스트용 데이터와 검증용 데이터를 사용해 모델 성능 측정
7. 모델 배포 : 모델이 선택되면 추론을 위해 제품으로 배포
8. 성능 모니터링 :
   - 모델 성능을 지속적으로 모니터링하고 그에 따라 재교육 및 보정.
   - 새로운 데이터를 수집해 계속해서 모델 개선
   - 오래된 데이터를 사용하지 않도록 주의

```
--[1] 'SCOTT'이 근무하는 부서명, 지역출력
--  .원하는 정보가 두 개 이상의 테이블에 나뉘어져 있을 때 결과출력.
```

```
select * from emp;
```

Status | Result1

	EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO
1	7369	SMITH	CLERK	7902	1980-12-1...	800	NULL	20
2	7499	ALLEN	SALE...	7698	1981-02-2...	1600	300	30
3	7521	WARD	SALE...	7698	1981-02-2...	1250	500	30
4	7566	JONES	MAN...	7839	1981-04-0...	2975	NULL	20
5	7654	MARTIN	SALE...	7698	1981-09-2...	1250	1400	30
6	7698	BLAKE	MAN...	7839	1981-05-0...	2850	NULL	30
7	7782	CLARK	MAN...	7839	1981-06-0...	2450	NULL	10
8	7788	SCOTT	ANA...	7566	1987-07-1...	3000	NULL	20
9	7839	KING	PRESI...	NULL	1981-11-1...	5000	NULL	10
10	7844	TURNER	SALE...	7698	1981-09-0...	1500	0	30
11	7876	ADAMS	CLERK	7788	1987-07-1...	1100	NULL	20
12	7900	JAMES	CLERK	7698	1981-12-0...	950	NULL	30
13	7902	FORD	ANA...	7566	1981-12-0...	3000	NULL	20
14	7934	MILLER	CLERK	7782	1982-01-2...	1300	NULL	10

```
select * from dept;
```

Status | Result1

	DEPTNO	DNAME	LOC
1	10	ACCOUNTING	NEW YORK
2	20	RESEARCH	DALLAS
3	30	SALES	CHICAGO
4	40	OPERATIONS	BOSTON

```
2 select deptno from emp
3 where ename = 'SCOTT';
```

Status | Result1

	DEPTNO
1	20

```
4
5 select dname, loc from dept
6 where deptno = 20;
```

	DNAME	LOC
1	RESEARCH	DALLAS