

<최종 보고서>

13011051 / 김준한

1. 요구사항 명세서

2020년 1학기 데이터베이스	요구사항 명세서	문서번호	1
		페이지번호	1/4 페이지
		작 성 자 (학번/이름)	13011051/김준 한
		작성일자	2020. 6.14.

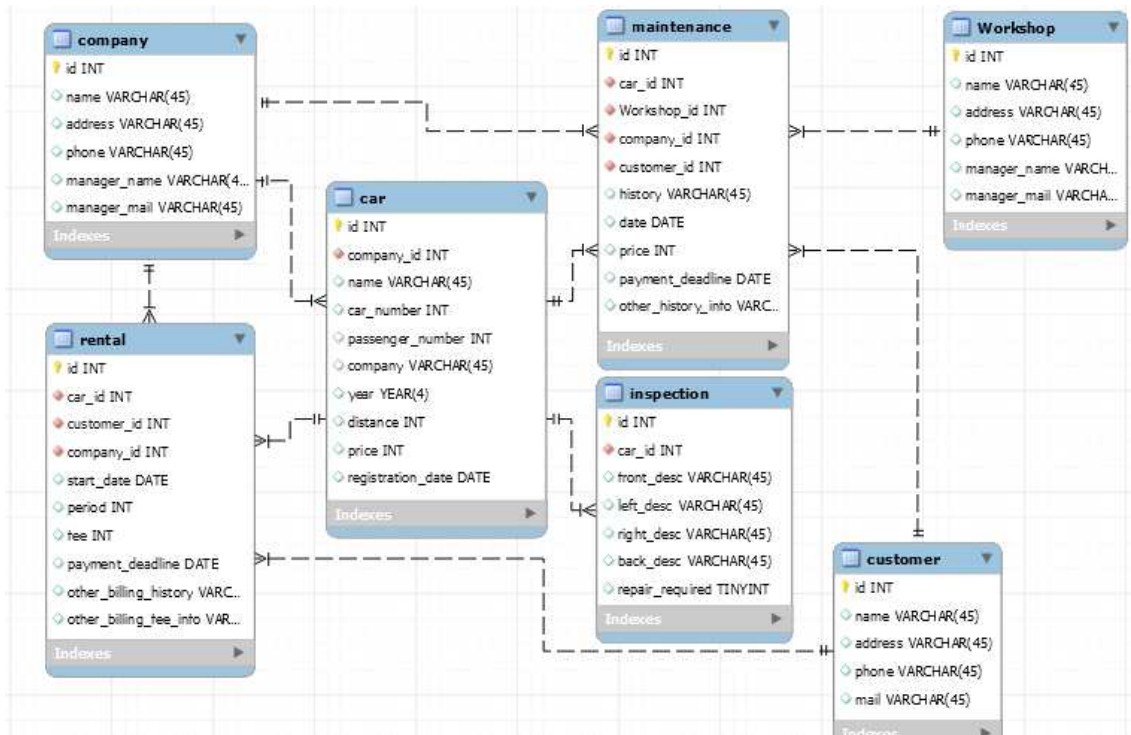
구 분	요 구 사 항	상 세 내 용	우선순위
1	스키마 정의 : company	(* 하단부터 별도 표기없을시 VARCHAR) 아이디(int), 이름, 주소, 전화번호, 관리자 이름, 관리자 이메일	1
2	스키마 정의 : car	아이디(int), 회사 아이디(외래키, int), 이름, 차량 번호(int), 승차 인원 수(int), 제조회사, 제조연도(y ear), 주행거리(int), 대여비용(int), 등록일자(date)	1
3	스키마 정의 : customer	아이디(int), 이름, 주소, 전화번호, 이메일	1
4	스키마 정의 : workshop	아이디(int), 이름, 주소, 전화번호, 관리자 이름, 관리자 메일	1
5	스키마 정의 : rental	아이디(int), 차량 아이디(외래키,int), 고객 아이디 (외래키,int), 회사 아이디(외래키int), 대여 일자(d ate), 대여 기간(int), 요금(int), 납입 기한(date), 기타청구내역, 기타청구요금정보	1
6	스키마 정의 : inspection	아이디(int), 차량 아이디(외래키,int), 전방 설명, 좌측 설명, 우측 설명, 후방 설명, 수리 필요 여부 (tinyint)	1

7	스키마 정의 : maintenance	아이디(int), 차량 아이디(외래키,int), 정비소 아이디(외래키,int), 회사 아이디(외래키,int), 고객 아이디(외래키,int), 정비 내역, 정비 일자(date), 정비 비용(int), 납입 기한(date), 기타 정비 내역 정보	1
8	스키마 정의 :스키마 연관 관계	company-car(1:n), company-rental(1:n), company-maintenance(1:n), car-rental(1:n), car-maintenance(1:n), car-inspection(1:n), customer-rental(1:n), customer-maintenance(1:n), workshop-maintenance(1:n)	1
9	질의어 : 초기화	“drop table if exists rental” ... “create table if not exists...” ... “on delete cascade...” ... “on update cascade...” ... “insert into [table] values... ” ...	2
10	질의어 : 입력	“insert into [company, car, customer, workshop, maintenance] values... ”	2
11	질의어 : 삭제	“delete from [company, car, customer, workshop, maintenance] where id = [id_value]”	2
12	질의어 : 변경	“update [company, car, customer, workshop, maintenance] set [column] = [column_value] where id = [id_value]” ...	2
13	질의어 : 캠핑카 검색	“select * from car where name=[name_value] and passenger_number between [passenger_number_min] and [passenger_number_max] and price between [price_min] and [price_max]”	2
14	질의어 : 캠핑카 대여	“insert into rental values... “	2
15	질의어 : 캠핑카 반환	“insert into inspection values... ”	2

16	질의어 : 검색 1	“select name from car where distance >= 80000 and id in (select car_id from maintenance where price >= 100000)”	2
17	질의어 : 검색 2	“select name from customer where id in (select customer_id from rental where car_id in (select id from car where passenger_number >= 5) group by customer_id having count(*) >= 1)”	2
18	질의어 : 검색 3	“select car.name from car, inspection where car.id=inspection.car_id and inspection.repair_required=0”	2
19	질의어 : 검색 4	“select name from company where (address=“서울“ or address=“경기“) and id in (select company_id from car where price >= 150000)”	2
20	인터페이스 : 관리자 / 사용자 영역	Border Layout 을 통해 영역을 분할하여, North에는 관리자, South에는 사용자 메뉴를 배치함.	3
21	인터페이스 : 출력 영역	Center에는 연산 결과, 에러 메시지 등을 출력하는 출력 창이 위치한다.	3
22	인터페이스 : 초기화	모든 테이블의 데이터가 초기화 된다.	3
23	인터페이스 : 대여 회사 입력	대여 회사 정보를 입력	3
24	인터페이스 : 캠핑카 입력	캠핑카 정보를 입력 (승차 인원 수 : 2~6, 대여 비용 : 50000~500000)	3
25	인터페이스 : 고객 입력	고객 정보를 입력	3

26	인터페이스 : 정비소 입력	정비소 정보를 입력	3
27	인터페이스 : [대여회사, 캠핑카, 고객, 정비소, 정비] 삭제	[대여회사, 캠핑카, 고객, 정비소, 정비] 정보를 cascade 옵션으로 삭제	3
28	인터페이스 : [대여회사, 캠핑카, 고객, 정비소, 정비] 변경	[대여회사, 캠핑카, 고객, 정비소, 정비] 정보를 cascade 옵션으로 변경. 입력된 id에 해당하는 튜플의 정보를 변경하며, 값을 입력한 것에 대해서만 변경한다.	3
29	인터페이스 : 캠핑카 검색	이름으로 검색, 승차 인원 수(2~6)으로 검색, 대여 비용으로 검색(50000~500000) 가능. 범위 검색에서 비워둔 값은 끝값으로 정해진다. 각 검색 조건을 동시에 적용할 수도 있다. ex) passenger_num : 5~6, price : 150000 ~	3
30	인터페이스 : 캠핑카 대여	차량 아이디, 고객 아이디, 대여 기간(2~6), 기타 청구 요금 정보(일시불, 3개월, 6개월)를 입력하여 캠핑카를 대여함. 출력단에는 반환 시에 이용할 수 있는 대여번호가 출력됨.	3
31	인터페이스 : 캠핑카 반환	대여 번호, 전방 설명, 좌측 설명, 우측 설명, 후방 설명, 수리 필요 여부(T / F)를 입력하여 캠핑카를 반환함. 출력단에는 정비 시에 이용할 수 있는 점검 번호(=대여 번호)가 출력됨.	3
32	인터페이스 : 캠핑카 정비 입력	점검 번호, 정비소 아이디, 정비 내역, 정비 일자, 정비 비용, 납입 기한을 입력하여 정비 정보를 입력함. 점검 시에 수리 필요 여부가 F 였다면 정비가 불가능하다. 수리 필요 여부가 T 였다면 정비 이후 F로 바뀌게 된다.	3
33	인터페이스 : 검색	번호(1 ~ 4)를 입력하면 해당 번호에 해당하는 검색 결과가 반환 된다. 1. 누적주행거리가 8만 이상이면서, 점검 가격이 한번이라도 10만원 이상이 나왔던 차량의 이름 2. 5인승이상의 차량을 1번이상 대여한 고객의 이름 3. 현재까지의 점검 목록(점검 테이블) 중 수리가 필요없는 차량의 이름 4. 서울,경기 소재 회사 중 대여비용이 15만원이상인 차량을 한대라도 보유한 회사의 이름	3

2. 스키마 정의



3. 프로그램 구조

CompanyInputDialog : 회사 입력
CarInputDialog : 차량 입력
CustomerInputDialog : 고객 입력
WorkshopInputDialog : 정비소 입력
DeleteDialog : 회사, 차량, 고객, 정비소, 정비 삭제
CompanyUpdateDialog : 회사 변경
CarUpdateDialog : 차량 변경
CustomerUpdateDialog : 고객 변경
WorkshopUpdateDialog : 정비소 변경
CarSearchDialog : 차량 검색
CarRentalDialog : 차량 대여
CarInspectionDialog : 차량 반환
CarMaintenanceInputDialog : 차량 정비 입력
CarMaintenanceUpdateDialog : 차량 정비 변경
SearchDialog : 검색 (4가지)
BookListSwing : 인터페이스 초기화, DB 연결, 스키마 생성, 데이터 초기화

4. 테스트 방법

● 검색을 제외한 테스트 결과는 **mysql workbench**를 통하여 별도로 확인한다.

1) 초기화 : 프로그램을 구동 후, 초기화를 눌러주어 스키마 및 데이터를 초기화 시켜준다.

초기화가 완료되면 init completed 메시지가 출력단에 출력.

2) 대여회사 입력 : (16, 세종, 서울, 01012345678, 김준한, kju@gmail.com) 입력

3) 캠핑카 입력 : (16, 4, 마티즈, 1234, 4, 대우, 1995, 50000, 100000, 2018-07-21) 입력

4) 고객 입력 : (16, 김준한, 서울, 01012345678, kju@gmail.com) 입력

5) 정비소 입력 : (16, 세종정비, 서울, 01012345678, 김준한, kju@gmail.com) 입력

6) 대여회사, 캠핑카, 고객, 정비소, 캠핑카 정비 삭제 :

해당 테이블에 존재하는 튜플의 id를 아무거나 입력.

cascade이므로 외래키로 인해 다른 테이블 항목도 삭제될 수 있음.

헛갈릴 수 있으니 한번의 삭제 테스트마다 초기화 해줄 것.

7) 대여회사, 캠핑카, 고객, 정비소, 캠핑카 정비 변경 :

해당 테이블에 존재하는 튜플의 id를 아무거나 입력 후,

그 외 항목들은 입력 예시를 적당히 수정하여 입력.

값을 비우면 해당 항목에 대해서는 변경되지 않음. 값이 채워진 항목만 변경됨.

cascade이므로 외래키로 인해 다른 테이블 항목도 변경될 수 있음.

헛갈릴 수 있으니 한번의 변경 테스트마다 초기화 해줄 것.

8) 대여 관련 테스트 진행을 위하여 다시 전체 초기화

(초기화 안하면 추가된 항목들에 의해서 아래의 테스트 결과가 달라질 수 있음)

9) 캠핑카 검색 : 값을 모두 비우고 검색 -> 15 개의 튜플들이 출력

passenger_number의 최소값을 5 -> id가 3,6,8,10,12,14 인 튜플들이 출력

price의 최대값을 100000 -> id : 2,7,8

passenger_number가 4~5, price가 100000~150000 -> id : 1,4,9

name이 티구안 -> id : 8

10) 캠핑카 대여 : (5, 8, 3, 일시불) 입력.

기존 rental 테이블의 마지막 아이디가 17이므로, 대여번호 18번이 부여됨.

11) 캠핑카 반환 : (18, 파손, 이상 없음, 이상 없음, 파손, T) 입력.

기존 대여 번호와 대응되는 점검(반환) 번호 18번이 부여됨.

inspection 테이블에 id가 18번이고, repair_required가 1인 튜플이 생성됨.

12) 캠핑카 정비 입력 : (18, 4, 수리, 2020-06-15, 50000, 2020-06-16) 입력.

inspection 테이블에 id가 18번인 튜플의 repair_required가 0이 됨.

13) 캠핑카 정비 입력 : inspection 테이블에 id가 12번인 튜플은 repair_required가 0임.

(12, 10, 수리, 2020-05-12, 73000, 2020-05-15) 입력.

No Repair Required for this Inspection ID 라고 뜨면 성공.

14) 다시 초기화

15) 검색 : 1 입력 -> k9, 스타렉스

2 입력 -> 김경훈, 송오영, 이민영, 민영화, 이준기

3 입력 -> 소나타, 스타렉스

4 입력 -> 리버, 다운