|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **훈련과정** | AWS 클라우드 기반 자바&스프링&리액트 풀스택 개발자 양성 | | | | | | | **훈련기간** | | 2025-02-10 ~ 2024-08-07(5회차) | | | | |
| **평가일시** | 2025. 04. 18 | | | **훈련생** | (인) | | | | **훈련교사** | | | | 허 범 회 (인) | |
| **교과목명** | | | | **평가방법** | | | | | **배점** | | | **점수** | **총점** | **성취**  **수준** |
| 데이터베이스 (Database) | | | | 수행  평가 | **서술형시험** | | | | 100 | | |  |  |  |
| **성취수준** | | 매우 우수(5수준) | | 우수(4수준) | | | 보통(3수준) | | 미흡(2수준) | | | | 매우 미흡(1수준) | |
| **성취기준** | | 90-100점 | | 80-89점 | | | 70-79점 | | 60-69점 | | | | 59점 이하 | |
| < 정기 평가의 목적 > | | | | | | | | | | | | | | |
| 1. 내실 있는 교육서비스의 제공  - 강의계획과 목표에 맞는 수업이 이뤄지고 있는지를 훈련생의 평가를 통한 검증  2. 객관적 평가를 통한 훈련생 개개인의 학업 성취도 파악 조치  - 학습저조자와 성적우수자에 대한 객관적인 판단  - 학습저조자의 경우 중도탈락을 방지하기 위한 사전 조치(보충학습 or 개별 과제 등)  3. 훈련 종료 후 훈련생 개인별 과학적이고 효율적인 학습계획 수립의 지표로 활용  - 훈련 종료 후 각 교과목별 평가 결과에 따라 취약과목과 강한 과목을 객관적으로 판단  - 시급하게 학습해야 할 과목 등 학습 계획 수립할 수 있도록 평가 지표로 활용 | | | | | | | | | | | | | | |
| < 평가자 유의사항 > | | | | | | | | | | | | | | |
| 1. 평가지는 모두 7**페이지**로 구성됐습니다. 평가지가 맞는지 평가 시작 전 반드시 확인하시기 바랍니다.  2. 평가지에 훈련생명을 반드시 기재하고, 서명해주십시오.  3. **평가 시작을 알리기 전까지 다음 장으로 넘기지 마십시오.**  4. 각 평가 문항별로 제시된 지시사항을 꼼꼼히 살피고, 지시사항에 맞게 평가해주십시오.  5. 각 평가 문항별로 채점기준을 꼼꼼히 살피고, 감점되지 않도록 주의하십시오.  6. **평가시간은 4시간**입니다. 평가 시간에 유의하십시오.  평가가 시작되기 전에 다음 장으로 넘기지 마십시오. | | | | | | | | | | | | | | |
| 평가방법 | | | | **서술형시험** | | | 배점 | | | | | 차등배점 | | | |
| ■ 평가문항(수행내용)  ※채점기준 : **SQL 구문을 직접 작성하여 기입하시오(또는 캡처)** 테이블 생성 CRUD 문제 다음과 같이 테이블을 정의 하시오 (10점)    **create table worker**  **(num int auto\_increment primary key,**  **id char(20),**  **name char(10),**  **gender char(1),**  **age int,**  **department char(20),**  **phone char(20),**  **address char(100));**    **create table salary (**  **salary\_id int auto\_increment primary key,**  **worker\_id char(20),**  **base\_salary int,**  **bonus int);**  2. worker 테이블과 , salary 테이블에 아래 데이터를 삽입하세요(10점)    **insert into worker (id, name, gender, age, department, phone, address) values (**  **'chkim', '김창훈','M',28,'총무부','010-3838-8577','경기도 용인시'),**  **('yckim', '김영철','M',32,'인사부','010-2222-3334','경기도 용인시'),**  **('yerin99', '김예린','F',25,'총무부','010-7777-9999','서울시 강서구'),**  **('seoklee', '이석훈','M',35,'영업부','010-8888-1111','서울시 마포구'),**  **('minseo88', '박민서','F',29,'기획부','010-9999-2222','경기도 성남시'),**  **('hjun45', '홍준표','M',41,'인사부','010-3333-4444','서울시 동작구'),**  **('suekim77', '김수애','F',27,'마케팅부','010-1212-3434','부산시 해운대구'),**  **('jkwon09', '권지민','F',30,'개발부','010-5656-7878','대전시 유성구');**    insert into salary (worker\_id, base\_salary, bonus) values  ('chkim', 3000000, 300000),  ('yckim', 2800000, 250000),  ('yerin99', 2700000, 200000),  ('seoklee', 3500000, 400000),  ('minseo88', 3200000, 300000),  ('hjun45', 4000000, 500000),  ('suekim77', 2900000, 200000),  ('jkwon09', 3300000, 350000);  3. ‘인사부’ 소속 사원들의 주소를 출력하시오.(3점)  SELECT name, department, address FROM worker where department='인사부';  4. ‘yerin99’ 사원의 보너스를 500000으로 수정하시오.(3점)  update salary set bonus=500000 where worker\_id='yerin99';  5. 총무부 사원의 급여 총액(기본급+보너스)을 출력하시오.(3점)  INNER JOIN  SELECT name, department, (base\_salary+bonus) as 급여총액 FROM worker INNER JOIN salary ON worker.id=salary.worker\_id WHERE department='총무부';  NON ANSI  SELECT name, department, (base\_salary+bonus) as 급여총액 FROM worker, salary WHERE worker.id=salary.worker\_id AND department='총무부'; | | | | | | | | | | | | | | | |
| 평가방법 | | | | **서술형시험** | | | 배점 | | | | | 차등배점 | | | |
| 집계함수 문제(총 21점) 6. WORKER 테이블에서 직원들의 전체 평균 나이를 구하세요. (3점)  SELECT AVG(age) as 전체평균나이 FROM worker;  7. 모든 직원들의 총 기본급(base\_salary) 합계를 구하세요. (3점)  SELECT SUM(base\_salary) AS 기본급합계 FROM salary;  8. 부서별(department) 직원 수를 구하세요. (3점)  SELECT department, count(\*) as 직원수 FROM worker group by department;  9. 보너스를 가장 많이 받은 금액을 출력하세요. (3점)  SELECT MAX(bonus) as 가장많은보너스금액 FROM salary;  10. 기본급이 가장 낮은 직원의 급여를 구하세요. (3점)  SELECT min(base\_salary) as 가장낮은기본급 FROM salary;  11. 부서별로 평균 나이가 30세 이상인 부서만 출력하세요. (3점)  SELECT department, avg(age) as 평균나이 FROM worker GROUP BY department HAVING avg(age)>=30;  12. 급여(salary 테이블)에서 기본급 + 보너스를 합산한 총액을 구하세요. (3점)  SELECT sum(base\_salary+bonus) as 기본급보너스총액 FROM salary;  JOIN 문제(총20점) 13. 사원의 이름과 급여 총액(기본급 + 보너스)를 조회하시오. (5점)  INNER JOIN  SELECT name, (base\_salary+bonus) as 급여총액 FROM worker INNER JOIN salary ON worker.id = salary.worker\_id;  NON ANSI  SELECT name, (base\_salary+bonus) as 급여총액 FROM worker, salary WHERE worker.id = salary.worker\_id;  14. 경기도에 거주하는 사원의 이름, 부서, 급여 총액을 출력하시오. (5점)  INNER JOIN  SELECT name, department, (base\_salary+bonus), address as 급여총액 FROM worker INNER JOIN salary ON worker.id = salary.worker\_id WHERE address like '경기도%';  NON ANSI  SELECT name, department, (base\_salary+bonus), address as 급여총액 FROM worker, salary WHERE worker.id = salary.worker\_id AND address like '경기도%';  15. 연구소 근무 사원의 이름과 기본급, 상여금을 조회하시오. (5점)  INNER JOIN  SELECT name, base\_salary, bonus FROM worker INNER JOIN salary ON worker.id = salary.worker\_id WHERE department like '%연구소%';  NON ANSI  SELECT name, base\_salary, bonus FROM worker, salary WHERE worker.id = salary.worker\_id AND department like '%연구소%';  16. 보너스가 300,000원 이상인 사원의 이름, 부서, 보너스를 출력하시오. (5점)  INNER JOIN  SELECT name, department, bonus FROM worker INNER JOIN salary ON worker.id = salary.worker\_id WHERE bonus>=300000;  NON ANSI  SELECT name, department, bonus FROM worker, salary WHERE worker.id = salary.worker\_id AND bonus>=300000; SQL 사용자 생성 / 데이터베이스 생성 / 삭제 문제(총 30점) 17. 사용자(User) 생성(5점)  관리자로 로그인한 상태에서, 아래 조건에 맞는 사용자를 생성하는 SQL을 작성하세요.  ✅ 사용자 ID : testuser ✅ 비밀번호 : Test1234!  create user testuser identified by 'Test1234!';  18. 사용자에게 권한 부여(5점)  grant all on \*.\* to testuser;  19. 17번 문제에서 생성한 testuser 계정에 아래 권한을 부여하세요. (5점)  sampledb라는 데이터베이스의 조회(SELECT) 와 데이터 삽입(INSERT) 권한을 부여하는 SQL을 작성하세요.  grant select, insert on sampledb.\* to testuser;  20. 데이터베이스 생성(5점)  새로운 프로젝트용 데이터베이스 project\_db를 생성하는 SQL을 작성하세요.  create database project\_db;  21. 사용자 삭제(5점)  testuser라는 이름의 사용자를 삭제하는 SQL을 작성하세요.  drop user testuser;  22. 데이터베이스 삭제(5점)  project\_db 데이터베이스를 삭제하는 SQL을 작성하세요.  drop database project\_db; | | | | | | | | | | | | | | | |
| 데이터베이스 문제해결형 평가 차등 배점표  |  |  |  |  |  | | --- | --- | --- | --- | --- | | 문항 | 평가 항목 | 만점 | 평가 기준 | 배점 사례 | | 1 | 테이블 생성 | 10점 | 테이블명, 컬럼명, 자료형까지 모두 정확 | 10점 / 8점 / 5점 / 0점 | | 2 | 데이터 삽입 | 10점 | INSERT 구문 정확 + 값 정확 | 10점 / 8점 / 5점 / 0점 | | 3 | 인사부 사원 주소 조회 | 3점 | WHERE 조건, SELECT 컬럼 정확 | 3점 / 2점 / 0점 | | 4 | yerin99 보너스 수정 | 3점 | UPDATE, WHERE 정확 | 3점 / 2점 / 0점 | | 5 | 총무부 사원 급여 총액 조회 | 3점 | SUM 함수 정확, WHERE 조건 정확 | 3점 / 2점 / 0점 | | 6 | 평균 나이 구하기 | 3점 | AVG 함수 정확 | 3점 / 2점 / 0점 | | 7 | 총 기본급 합계 구하기 | 3점 | SUM 함수 정확 | 3점 / 2점 / 0점 | | 8 | 부서별 직원 수 구하기 | 3점 | GROUP BY + COUNT 사용 정확 | 3점 / 2점 / 0점 | | 9 | 보너스 최대 금액 조회 | 3점 | MAX 함수 사용 정확 | 3점 / 2점 / 0점 | | 10 | 기본급 최저 급여 조회 | 3점 | MIN 함수 사용 정확 | 3점 / 2점 / 0점 | | 11 | 평균 나이 30세 이상 부서 조회 | 3점 | HAVING 절 포함 GROUP BY 정확 | 3점 / 2점 / 0점 | | 12 | 기본급+보너스 총액 계산 | 3점 | SELECT 식 정확 | 3점 / 2점 / 0점 | | 13 | 이름+급여총액 JOIN 조회 | 5점 | JOIN 정확 + SELECT 정확 | 5점 / 4점 / 3점 / 0점 | | 14 | 경기도 거주 사원 정보 조회 | 5점 | JOIN + WHERE 조건 정확 | 5점 / 4점 / 3점 / 0점 | | 15 | 연구소 사원 정보 조회 | 5점 | JOIN + 조건 정확 | 5점 / 4점 / 3점 / 0점 | | 16 | 보너스 30만 이상 사원 조회 | 5점 | JOIN + WHERE 조건 정확 | 5점 / 4점 / 3점 / 0점 | | 17 | 사용자 생성 | 5점 | CREATE USER 문법 정확 | 5점 / 3점 / 0점 | | 18 | 사용자 권한 부여 | 5점 | GRANT 문법 정확 | 5점 / 3점 / 0점 | | 19 | DB 권한 부여 | 5점 | GRANT SELECT, INSERT 정확 | 5점 / 3점 / 0점 | | 20 | 데이터베이스 생성 | 5점 | CREATE DATABASE 문법 정확 | 5점 / 3점 / 0점 | | 21 | 사용자 삭제 | 5점 | DROP USER 문법 정확 | 5점 / 3점 / 0점 | | 22 | 데이터베이스 삭제 | 5점 | DROP DATABASE 문법 정확 | 5점 / 3점 / 0점 | | | | | | | | | | | | | | | | |