

교재명	44개 출제포인트로 4주만에 합격하는 2025 이패스 SQLD
작성일	2025.8.21

페이지	정오항목	수정 전	수정 후						
48	3. 관계의 표기법	<p>3. 관계의 표기법</p> <ul style="list-style-type: none"> • 관계명 : 관계의 이름 <ul style="list-style-type: none"> - 관계에 엔티티가 참여하는 형태를 정의한다. - 각 두개의 관계명을 가진다. - 관계명은 능동적 또는 수동적으로 명명된다. - 애매한 동사를 피하고 현재형으로 표현한다. • 관계차수 : 1:1, 1:M, M:M <ul style="list-style-type: none"> - 관계있는 엔티티간에 참여하는 수를 의미한다. • 관계선택사항 : 필수관계, 선택관계 <ul style="list-style-type: none"> - 엔티티가 관계에 있어 항상 참여하는지, 선택적 참여를 하는지 표기한다. 	M:N						
49	3. 관계의 표기법		삭제						
59	4번 문제	<p>2. 이상현상(Abnormality)</p> <p>데이터 모델링에서 이상현상(Abnormality)은 데이터베이스 설계와 운영 중 발생할 수 있는 비정상적인 동작이나 문제를 의미한다. 주로 데이터베이스가 정규화되지 않았거나 잘못된 설계로 인해 발생한다. 이상현상은 다음과 같이 세 가지 주요 유형으로 나뉜다.</p> <table border="1"> <tr> <td>삽입이상</td> <td>행(Row) 삽입 시 지정되지 않은 속성 값이 NULL을 가지는 경우</td> </tr> <tr> <td>갱신이상</td> <td>데이터 갱신 시 일부분의 데이터만 갱신되어 일관성 유지가 안되는 경우</td> </tr> <tr> <td>삭제이상</td> <td>행(Row) 삭제 시 원하지 않는 연쇄 삭제가 발생하는 현상</td> </tr> </table>	삽입이상	행(Row) 삽입 시 지정되지 않은 속성 값이 NULL을 가지는 경우	갱신이상	데이터 갱신 시 일부분의 데이터만 갱신되어 일관성 유지가 안되는 경우	삭제이상	행(Row) 삭제 시 원하지 않는 연쇄 삭제가 발생하는 현상	이상현상(Abnormality)은
삽입이상	행(Row) 삽입 시 지정되지 않은 속성 값이 NULL을 가지는 경우								
갱신이상	데이터 갱신 시 일부분의 데이터만 갱신되어 일관성 유지가 안되는 경우								
삭제이상	행(Row) 삭제 시 원하지 않는 연쇄 삭제가 발생하는 현상								
79	2번 문제	<p>12 다음 중 ANSI-SPARC에서 정의한 3단계구조(three-level architecture)에서 아래 내용이 설명하는 스키마구조로 가장 적절한 것은?</p> <ul style="list-style-type: none"> • 모든 사용자 관점을 통합한 조직 전체 관점의 통합적 표현 • 모든 응용시스템들이나 사용자들이 필요로 하는 데이터를 통합한 조직 전체의 DB를 기술할 것으로, DB에 저장되는 데이터와 그들간의 관계를 표현하는 스키마 <p>① 외부스키마(External Schema) ② 개념스키마(Conceptual Schema) ③ 내부스키마(Internal Schema) ④ 논리스키마(Logical Schema)</p> <p>해설 데이터베이스 스키마 구조는 3단계로 구분되며, 각각은 상호독립적인 의미와 고유한 기능을 가진다. 그 중 통합 관점의 스키마를 개념스키마(Conceptual Schema)라 한다.</p>	Schema						
83	9번 문제	<pre>SELECT 건급사건번호, 사건명 FROM 건급사건 WHERE 발생일시 = '20150905' UNION ALL SELECT 특수사건번호, 사건명 FROM 특수사건 WHERE 발생일시 = '20150905' UNION ALL SELECT 일반사건번호, 사건명 FROM 일반사건 WHERE 발생일시 = '20150905' UNION ALL</pre> <p>* 전체조건 : 위 세 테이블은 함께 조회하는 경우가 대부분이고 아직 시스템을 오픈하지 않았다.</p> <p>① UNION ALL로 조회하면 정렬로 인한 상충이 사라지므로 UNION으로 조회하여 조회한다. ② 건급사건, 특수사건, 일반사건을 하나의 테이블로 통합하고 PK를 사건분류코드에 결합 호로 조합하여 구성하도록 한다. ③ 건급사건, 특수사건, 일반사건 테이블을 개별로 유지하되 PK에 사건분류코드를 포함하도록 한다. ④ 세 개의 테이블을 그대로 유지하되 반정규화된 형태의 통합테이블을 하나 더 생성하여 조회의 성능을 향상하도록 한다.</p> <p>해설 개별 테이블을 모두 조회하는 트랜잭션이 대부분이라는 가정이었으므로 UNION/UNION ALL일 경우 개별 조회에 따른 시간소모와 이점을 조합하는 성능저하가 발생한다. 따라서 하나의 테이블로 통합하도록 하고 대신 PK제거나 일관성에서 각 사건을 구분 할 수 있도록 구분자를 부여한다.</p>	사건변						
102	5. NVL2	<p>5. NVL2</p> <ul style="list-style-type: none"> • NVL2는 NVL에 비해 파라미터가 하나 더 있다. NVL2(값, 'null이 아닐 때 출력 값', 'null일 때 출력값') <pre>select id, name, nvl(bonus_type, '배당없음') as NVL_, nvl2(bonus, '보너스 대상자', '미 대상자') as NVL2_ from salary</pre>	NVL2						

122	1. 서브쿼리란	<ul style="list-style-type: none"> 서브쿼리는 시작과 끝을 괄호로 감싸서 표현한다. 서브쿼리는 사용되는 위치에 따라 단일행으로 출력되거나 복수행으로 출력 될 수 있다. <table border="1"> <tr> <td>단일행 서브쿼리</td> <td>[< , > , <= > 사용가능</td> </tr> <tr> <td>다중행 서브쿼리</td> <td>IN, ALL, ANY, EXISTS 사용가능</td> </tr> <tr> <td>CROSS JOIN</td> <td>JOIN 하는 테이블간 조건이 없을 때 실행되는 JOIN</td> </tr> <tr> <td>OUTER JOIN</td> <td>INNER JOIN과 같은 상태에서 기준이 되는 테이블은 다른 쪽에 같은 값이 없어도 실행되어 출력하는 JOIN</td> </tr> </table>	단일행 서브쿼리	[< , > , <= > 사용가능	다중행 서브쿼리	IN, ALL, ANY, EXISTS 사용가능	CROSS JOIN	JOIN 하는 테이블간 조건이 없을 때 실행되는 JOIN	OUTER JOIN	INNER JOIN과 같은 상태에서 기준이 되는 테이블은 다른 쪽에 같은 값이 없어도 실행되어 출력하는 JOIN	복수행으로																																																																								
단일행 서브쿼리	[< , > , <= > 사용가능																																																																																		
다중행 서브쿼리	IN, ALL, ANY, EXISTS 사용가능																																																																																		
CROSS JOIN	JOIN 하는 테이블간 조건이 없을 때 실행되는 JOIN																																																																																		
OUTER JOIN	INNER JOIN과 같은 상태에서 기준이 되는 테이블은 다른 쪽에 같은 값이 없어도 실행되어 출력하는 JOIN																																																																																		
132	4번 정답 및 해설	<p>정답 도해상</p> <p>01 O UNION 연산은 기본적으로 중복을 제거한 결과를 반환한다. 02 O UNION 연산은 결과 집합에서 중복된 행을 제거한다. 03 O UNION ALL은 중복된 행을 포함한다. 04 O UNION 연산의 결과는 기본적으로 정렬됩니다. 05 X UNION ALL은 중복을 제거하지 않는다. 06 O UNION은 결과를 결합하고 중복을 제거한다. 07 X UNION은 중복을 제거하고, UNION ALL은 중복을 유지한다.</p>	X UNION 연산의 결과는 기본적으로 정렬되지 않고, ORDER BY 절을 통해 별도의 정렬이 필요합니다.																																																																																
143	4. PERCENT_RANK	<p>4. PERCENT_RANK</p> <ul style="list-style-type: none"> PERCENT_RANK는 SQL에서 윈도우 함수의 일종으로, 특정 행의 순위를 백분율로 계산하는 데 사용된다. 데이터 집합 내에서 각 행이 차지하는 상대적인 위치를 나타내며, 0과 1 사이의 값을 반환한다. PERCENT_RANK는 다음과 같은 공식용 사용하여 계산된다. $\text{PERCENT_RANK} = \frac{\text{rank} - 1}{\text{ROWSTOTAL} - 1}$	$\text{ERCENT RANK} = \frac{\text{rank} - 1}{\text{ROWSTOTAL} - 1}$																																																																																
157	3. 정규표현식에서 자주 쓰이는 패턴	<p>3. 정규표현식에서 자주 쓰이는 패턴</p> <table border="1"> <thead> <tr> <th>패턴</th> <th>설명</th> <th>예제</th> <th>결과</th> </tr> </thead> <tbody> <tr> <td>.</td> <td>임의의 한 문자(자음 제외)</td> <td>'abc'</td> <td>abc, abc</td> </tr> <tr> <td>*</td> <td>문자열의 시작</td> <td>'*abc'</td> <td>abc, abc1</td> </tr> <tr> <td>\$</td> <td>문자열의 끝</td> <td>'abc\$'</td> <td>abc, abc</td> </tr> <tr> <td>\d</td> <td>숫자(0-9)</td> <td>'\d*'</td> <td>123, 4567</td> </tr> <tr> <td>\D</td> <td>숫자가 아닌 문자</td> <td>'\D*'</td> <td>abc, hello</td> </tr> <tr> <td>\w</td> <td>알파벳, 숫자, 밑줄(=[a-zA-Z0-9_])</td> <td>'\w*'</td> <td>abc123, hello_1</td> </tr> <tr> <td>\W</td> <td>알파벳, 숫자, 밑줄이 아닌 문자</td> <td>'\W*'</td> <td>@#%, *!@</td> </tr> </tbody> </table>	패턴	설명	예제	결과	.	임의의 한 문자(자음 제외)	'abc'	abc, abc	*	문자열의 시작	'*abc'	abc, abc1	\$	문자열의 끝	'abc\$'	abc, abc	\d	숫자(0-9)	'\d*'	123, 4567	\D	숫자가 아닌 문자	'\D*'	abc, hello	\w	알파벳, 숫자, 밑줄(=[a-zA-Z0-9_])	'\w*'	abc123, hello_1	\W	알파벳, 숫자, 밑줄이 아닌 문자	'\W*'	@#%, *!@	삭제																																																
패턴	설명	예제	결과																																																																																
.	임의의 한 문자(자음 제외)	'abc'	abc, abc																																																																																
*	문자열의 시작	'*abc'	abc, abc1																																																																																
\$	문자열의 끝	'abc\$'	abc, abc																																																																																
\d	숫자(0-9)	'\d*'	123, 4567																																																																																
\D	숫자가 아닌 문자	'\D*'	abc, hello																																																																																
\w	알파벳, 숫자, 밑줄(=[a-zA-Z0-9_])	'\w*'	abc123, hello_1																																																																																
\W	알파벳, 숫자, 밑줄이 아닌 문자	'\W*'	@#%, *!@																																																																																
190	5번 문제	<p>05 아래의 테이블 스키마 정보를 참고하여, 다음 중 '구매이력'이 있는 고객 중 구매 횟수가 3회 이상인 고객의 이름과 등급을 출력하시오. 라는 질의에 대해 아래 SQL 문장의 () , ()에 들어갈 구문으로 가장 적절한 것은?</p> <pre> [테이블] 고객(고객번호(PK), 이름, 등급) 구매이력(구매번호(PK), 구매금액, 고객번호(PK)) * 구매이력(테이블의 고객번호는 고객 테이블의 고객 번호를 참조하는 외래키 Foreign Key) [SQL 문장] SELECT A.이름, A.등급 FROM 고객 A () () ORDER BY A.이름, A.등급 () </pre>	Key Group																																																																																
275	34번 문제	<p>34 아래의 SQL 중 결과가 다른 것은?</p> <p>① LPAD('BCD','A',3) ② RTRIM('EBCDE','E') ③ SUBSTR('ABCDE',2,3) ④ CONCAT(CONCAT(NULL,'BC'),'E')</p>	'D'																																																																																
278	28번 문제 정답	<p>과목 II SQL 기본 및 활용</p> <table border="1"> <tr> <td>01</td><td>③</td><td>02</td><td>③</td><td>03</td><td>①,③</td><td>04</td><td>③</td><td>05</td><td>①</td> </tr> <tr> <td>06</td><td>②</td><td>07</td><td>②,④</td><td>08</td><td>④</td><td>09</td><td>②</td><td>10</td><td>④</td> </tr> <tr> <td>11</td><td>②</td><td>12</td><td>④</td><td>13</td><td>③</td><td>14</td><td>①,③</td><td>15</td><td>③</td> </tr> <tr> <td>16</td><td>③</td><td>17</td><td>③</td><td>18</td><td>②</td><td>19</td><td>②</td><td>20</td><td>①</td> </tr> <tr> <td>21</td><td>①</td><td>22</td><td>③</td><td>23</td><td>③</td><td>24</td><td>②</td><td>25</td><td>④</td> </tr> <tr> <td>26</td><td>②</td><td>27</td><td>③</td><td>28</td><td>③</td><td>29</td><td>①,③</td><td>30</td><td>③</td> </tr> <tr> <td>31</td><td>④</td><td>32</td><td>③</td><td>33</td><td>②</td><td>34</td><td>①,③</td><td>35</td><td>④</td> </tr> <tr> <td>36</td><td>①</td><td>37</td><td>①</td><td>38</td><td>④</td><td>39</td><td>②</td><td>40</td><td>③</td> </tr> </table>	01	③	02	③	03	①,③	04	③	05	①	06	②	07	②,④	08	④	09	②	10	④	11	②	12	④	13	③	14	①,③	15	③	16	③	17	③	18	②	19	②	20	①	21	①	22	③	23	③	24	②	25	④	26	②	27	③	28	③	29	①,③	30	③	31	④	32	③	33	②	34	①,③	35	④	36	①	37	①	38	④	39	②	40	③	②
01	③	02	③	03	①,③	04	③	05	①																																																																										
06	②	07	②,④	08	④	09	②	10	④																																																																										
11	②	12	④	13	③	14	①,③	15	③																																																																										
16	③	17	③	18	②	19	②	20	①																																																																										
21	①	22	③	23	③	24	②	25	④																																																																										
26	②	27	③	28	③	29	①,③	30	③																																																																										
31	④	32	③	33	②	34	①,③	35	④																																																																										
36	①	37	①	38	④	39	②	40	③																																																																										

280	28번 문제 정답	<p>28 ㉓</p> <p>GROUP BY 절의 집합을 원본으로 하는 데이터를 WINDOW FUNCTION과 함께 사용하면 GROUP BY 절과 함께 WINDOW FUNCTION을 사용한다고 하더라도 오류가 발생하지 않으며, 유사개수 컬럼은 상품분류 코드로 GROUPING된 집합을 원본집합으로 하여 상품분류코드별 평균상품가격을 서로 비교하여 현재 위치한 상품분류코드의 평균가격 대비 -1000 ~ +10000 사이에 존재하는 상품분류 코드의 개수를 구한 것이다.</p>	㉔
337	40 반정규화	<ul style="list-style-type: none"> • 종류 - 테이블 병합 1:1 / 1:M - 수퍼 / 서브타입 병합 - 부분데이터 병합 - 통계데이터 병합 - 중복데이터 병합 - 부분데이터 병합 - 이력 컬럼 추가 - 중복 컬럼 추가 - PK를 일반 컬럼으로 병합 - 파생 컬럼 추가 - 응용시스템 오직종을 피하기 위한 임시값 컬럼 추가 - 중복관계 추가 	삭제