

SQL Practice Questions

AutonomousCarDecisionSystem Database | 50 Questions

Easy → Very Hard | Includes Full Memo

DATABASE STRUCTURE — QUICK REMINDER

Actions Table

Field	Type	Description
action_id	AutoNumber	Unique ID for each action
action_name	Text	Name of the action (e.g. Brake, Swerve Left)
people_involved	Integer	Number of people affected by this action
action_cost	Integer	Cost score 1-10 (how damaging the action is)
legal_action	Boolean	True = legal, False = illegal

Danger Table

Field	Type	Description
Danger_id	AutoNumber	Unique ID for each danger
Danger_type	Text	Category (Pedestrian, Cyclist, Animal, Obstacle, Vehicle)
Danger_description	Text	Short description (e.g. Child, Adult cyclist, Dog)
people_count	Integer	Number of people involved (0 if none)
risk_weight	Integer	1 = low risk, 2 = medium risk, 3 = high risk
location_zone	Text	Where detected (School Zone, Residential, Highway, CBD)

Decision_link Table

Field	Type	Description
link_id	AutoNumber	Unique ID for each link record
action_id	Integer	Links to Actions.action_id
Danger_id	Integer	Links to Danger.Danger_id
recommended	Boolean	True = recommended, False = not recommended
priority_score	Integer	Urgency score 1-10
notes	Text	Reason/comment for the decision
incident_date	Date/Time	Date the incident was recorded

Difficulty Key

Level	Description	Expected clauses
Easy	Single table, basic SELECT/WHERE	SELECT, FROM, WHERE, ORDER BY
Medium	Aggregate functions, GROUP BY, HAVING	COUNT, AVG, SUM, MAX, MIN, GROUP BY, HAVING
Hard	Multiple tables, joins, string/date functions	Multi-table WHERE joins, LIKE, BETWEEN, date functions

Level	Description	Expected clauses
Very Hard	Subqueries, combined logic, action queries	Subqueries, INSERT, UPDATE, DELETE, complex WHERE

PART 1 — EASY QUESTIONS (Q1–Q15)

Q1 [Easy] Display all records from the Actions table.

Q2 [Easy] Display only the action_name and action_cost fields from the Actions table.

Q3 [Easy] Display all Danger records where the risk_weight is 3 (high risk).

Q4 [Easy] Display all actions where the action is legal (legal_action = True).

Q5 [Easy] Display all Danger records sorted by risk_weight from highest to lowest.

Q6 [Easy] Display all Danger records where the location_zone is 'Highway'.

Q7 [Easy] Display the first 3 action names from the Actions table in alphabetical order.

Q8 [Easy] Display all Decision_link records where recommended is False.

Q9 [Easy] Display all Danger records where the people_count is 0 (no people involved).

Q10 [Easy] Display all actions where the action_cost is less than 5, sorted by action_cost ascending.

Q11 [Easy] Display the Danger_type and location_zone for all Danger records where the Danger_type is 'Animal'.

Q12 [Easy] Display all Decision_link records where the priority_score is 10.

Q13 [Easy] Display all Danger records where the location_zone is either 'Residential' or 'CBD'.

Q14 [Easy] Display all Action records where the action_name contains the word 'Swerve'.

Q15 [Easy] Display all Danger records where the risk_weight is between 2 and 3 (inclusive).

PART 2 — MEDIUM QUESTIONS (Q16–Q30)

Q16 [Medium] Count the total number of records in the Danger table. Display the result under the heading TotalDangers.

Q17 [Medium] Show the highest priority_score in the Decision_link table. Display it under the heading MaxPriority.

Q18 [Medium] Calculate the average action_cost from the Actions table. Display it under the heading AvgCost.

Q19 [Medium] Show the total (SUM) of all people_count values in the Danger table. Display it as TotalPeople.

Q20 [Medium] Show the number of Danger records for each location_zone. Display the location_zone and the count as NumDangers.

Q21 [Medium] Show the average priority_score for each recommended status (True/False) in the Decision_link table. Display recommended and the average as AvgPriority.

Q22 [Medium] Show only location zones that have more than 1 Danger record. Display the location_zone and count as NumDangers.

Q23 [Medium] Display the minimum and maximum action_cost from the Actions table. Label them MinCost and MaxCost.

Q24 [Medium] For each Danger_type, calculate the total risk as SUM of (people_count x risk_weight). Display Danger_type and the total as TotalRisk. Sort from highest to lowest TotalRisk.

Q25 [Medium] Display each Danger_type and the average risk_weight for that type. Only show types where the average risk_weight is greater than 1.5. Label the average AvgRisk.

Q26 [Medium] Create a DangerCode for each Danger by taking the first 3 letters of Danger_type and the last 2 letters of location_zone combined together. Display Danger_type, location_zone and DangerCode.

Q27 [Medium] Display the Danger_type, Danger_description and the number of characters in the Danger_description field. Label the count DescLength.

Q28 [Medium] For each action, display the action_name and a calculated field called AdjustedCost which is action_cost multiplied by people_involved. Sort by AdjustedCost descending.

Q29 [Medium] Display each Danger_type and the count of recommended actions linked to it. Only include recommended = True records. Label the count RecommendedCount. Sort by RecommendedCount descending.

Q30 [Medium] Display the Danger_type and the number of distinct location zones it appears in. Label this UniqueZones. Only show types with more than 1 unique zone.

PART 3 — HARD QUESTIONS (Q31–Q43)

Q31 [Hard] Display the `action_name` and `Danger_type` for all records where an action has been linked to a Danger. Use all three tables.

Q32 [Hard] Display the `action_name`, `Danger_type`, and `priority_score` for all recommended actions, sorted from highest `priority_score` to lowest.

Q33 [Hard] Display the `action_name`, `Danger_type`, `Danger_description` and `location_zone` of all linked records where the danger occurred in a School Zone.

Q34 [Hard] Display the `action_name`, `Danger_type` and notes for all decision links where the action is NOT recommended.

Q35 [Hard] Display the `action_name`, `Danger_type` and `priority_score` of linked records where the `priority_score` is between 8 and 10 inclusive.

Q36 [Hard] Display the `Danger_type`, `Danger_description`, and the YEAR, MONTH, and DAY of the `incident_date` for all `Decision_link` records. Label the columns `IncidentYear`, `IncidentMonth`, `IncidentDay`.

Q37 [Hard] Display all linked incidents recorded in the current year. Show `Danger_type`, `Danger_description`, `incident_date` and `location_zone`.

Q38 [Hard] Display the `action_name` and `Danger_type` for all linked records where the danger has a `risk_weight` of 3 AND the action is recommended.

Q39 [Hard] Display the `action_name`, `Danger_type` and average `priority_score` per `Danger_type` for all linked records. Label the average `AvgPriority`. Only show Danger types with an average priority score above 6.

Q40 [Hard] Display the `action_name` and the total number of Dangers linked to it. Label this `NumDangersLinked`. Only show actions linked to more than 3 different Dangers. Sort from highest to lowest.

Q41 [Hard] Display the `action_name`, `Danger_type` and `people_count` for all linked records where the `Danger_type` is 'Pedestrian' and the action is recommended. Sort by `people_count` descending.

Q42 [Hard] Create a `SimCode` for each action by combining the first 2 letters of `action_name` and the `action_id`. Display `action_name` and `SimCode`.

Q43 [Hard] Display the `action_name`, `Danger_type` and `priority_score` for all linked records where the notes field contains the word 'risk'. Sort by `priority_score` descending.

PART 4 — VERY HARD QUESTIONS (Q44–Q50)

Q44 [Very Hard] Display the Danger(s) with the highest risk_weight. Show Danger_type, Danger_description and risk_weight. Use a subquery.

Q45 [Very Hard] Display the names of actions that have NEVER been recommended for any Danger. Use a subquery with NOT IN.

Q46 [Very Hard] Display the recommended actions for the Danger with the highest total risk (people_count x risk_weight). Show action_name, Danger_type and priority_score sorted from highest to lowest. Use a subquery.

Q47 [Very Hard] Increase the priority_score by 2 for all Decision_link records where recommended = True AND the Danger has a risk_weight of 3. Write the UPDATE query.

Q48 [Very Hard] Insert a new Danger record with the following values: Danger_type = 'Vehicle', Danger_description = 'Speeding truck', people_count = 3, risk_weight = 3, location_zone = 'Highway'. Write the INSERT query.

Q49 [Very Hard] Delete all Decision_link records where the priority_score is less than 3 AND recommended = False. Write the DELETE query.

Q50 [Very Hard] Calculate the average risk_weight of Dangers that have MORE than 2 recommended actions linked to them. Use a subquery. Label the result AvgRiskOfPopularDangers.

MEMORANDUM — FULL ANSWERS

Each answer includes the SQL query and a brief explanation of key clauses used.

MEMO: PART 1 — EASY (Q1–Q15)

Q1 [Easy] *Display all records from the Actions table.*

```
SELECT *  
FROM Actions;
```

Mark: 1 mark — correct SELECT * and FROM Actions

Q2 [Easy] *Display only action_name and action_cost from Actions.*

```
SELECT action_name, action_cost  
FROM Actions;
```

Mark: 1 mark — both correct fields listed

Q3 [Easy] *All Danger records where risk_weight = 3.*

```
SELECT *  
FROM Danger  
WHERE risk_weight = 3;
```

Mark: 2 marks — correct table (1), correct WHERE condition (1)

Q4 [Easy] *All actions where legal_action = True.*

```
SELECT *  
FROM Actions  
WHERE legal_action = True;
```

Mark: 2 marks — correct table (1), correct boolean WHERE (1)

Q5 [Easy] *All Danger records sorted by risk_weight descending.*

```
SELECT *  
FROM Danger  
ORDER BY risk_weight DESC;
```

Mark: 2 marks — correct table (1), DESC sort (1)

Q6 [Easy] All Danger where location_zone = 'Highway'.

```
SELECT *  
FROM Danger  
WHERE location_zone = 'Highway';
```

Mark: 2 marks — correct WHERE field and value

Q7 [Easy] First 3 action names alphabetically.

```
SELECT TOP 3 action_name  
FROM Actions  
ORDER BY action_name;
```

Mark: 3 marks — TOP 3 (1), correct field (1), ORDER BY ASC (1)

Q8 [Easy] All Decision_link records where recommended = False.

```
SELECT *  
FROM Decision_link  
WHERE recommended = False;
```

Mark: 2 marks — correct table (1), boolean condition (1)

Q9 [Easy] All Danger where people_count = 0.

```
SELECT *  
FROM Danger  
WHERE people_count = 0;
```

Mark: 2 marks — correct table (1), WHERE condition (1)

Q10 [Easy] Actions where action_cost < 5, sorted ascending.

```
SELECT *  
FROM Actions  
WHERE action_cost < 5  
ORDER BY action_cost ASC;
```

Mark: 3 marks — correct WHERE (1), correct comparison (1), ASC sort (1)

Q11 [Easy] Danger_type and location_zone where Danger_type = 'Animal'.

```
SELECT Danger_type, location_zone  
FROM Danger  
WHERE Danger_type = 'Animal';
```

Mark: 3 marks — correct fields (1), correct table (1), correct WHERE (1)

Q12 [Easy] *Decision_link records where priority_score = 10.*

```
SELECT *  
FROM Decision_link  
WHERE priority_score = 10;
```

Mark: 2 marks — correct table (1), WHERE condition (1)

Q13 [Easy] *Danger where location_zone is Residential OR CBD.*

```
SELECT *  
FROM Danger  
WHERE location_zone = 'Residential' OR location_zone = 'CBD';
```

Alternative answer:

```
WHERE location_zone IN ('Residential', 'CBD');
```

Mark: 3 marks — correct table (1), both values included (1), OR / IN used correctly (1)

Q14 [Easy] *Actions where action_name contains 'Swerve'.*

```
SELECT *  
FROM Actions  
WHERE action_name LIKE '*Swerve*';
```

Mark: 3 marks — correct table (1), LIKE used (1), correct wildcard pattern (1)

Q15 [Easy] *Danger where risk_weight is between 2 and 3.*

```
SELECT *  
FROM Danger  
WHERE risk_weight BETWEEN 2 AND 3;
```

Mark: 3 marks — correct table (1), BETWEEN used (1), correct values (1)

MEMO: PART 2 — MEDIUM (Q16–Q30)

Q16 [Medium] Total number of records in Danger table as TotalDangers.

```
SELECT COUNT(*) AS TotalDangers
FROM Danger;
```

Mark: 2 marks — COUNT(*) (1), correct alias (1)

Q17 [Medium] Highest priority_score as MaxPriority.

```
SELECT MAX(priority_score) AS MaxPriority
FROM Decision_link;
```

Mark: 2 marks — MAX function (1), correct field and alias (1)

Q18 [Medium] Average action_cost as AvgCost.

```
SELECT AVG(action_cost) AS AvgCost
FROM Actions;
```

Mark: 2 marks — AVG function (1), correct alias (1)

Q19 [Medium] Total SUM of people_count as TotalPeople.

```
SELECT SUM(people_count) AS TotalPeople
FROM Danger;
```

Mark: 2 marks — SUM function (1), correct alias (1)

Q20 [Medium] Number of Danger records per location_zone as NumDangers.

```
SELECT location_zone, COUNT(*) AS NumDangers
FROM Danger
GROUP BY location_zone;
```

Mark: 3 marks — COUNT(*) (1), correct alias (1), GROUP BY location_zone (1)

Q21 [Medium] Average priority_score per recommended status as AvgPriority.

```
SELECT recommended, AVG(priority_score) AS AvgPriority
FROM Decision_link
GROUP BY recommended;
```

Mark: 3 marks — AVG (1), GROUP BY recommended (1), correct alias (1)

Q22 [Medium] *Location zones with more than 1 Danger record.*

```
SELECT location_zone, COUNT(*) AS NumDangers
FROM Danger
GROUP BY location_zone
HAVING COUNT(*) > 1;
```

Mark: 4 marks — COUNT (1), GROUP BY (1), HAVING (1), correct condition (1)

Q23 [Medium] *Minimum and maximum action_cost as MinCost and MaxCost.*

```
SELECT MIN(action_cost) AS MinCost, MAX(action_cost) AS MaxCost
FROM Actions;
```

Mark: 3 marks — MIN (1), MAX (1), both aliases correct (1)

Q24 [Medium] *Total risk per Danger_type sorted descending.*

```
SELECT Danger_type, SUM(people_count * risk_weight) AS TotalRisk
FROM Danger
GROUP BY Danger_type
ORDER BY TotalRisk DESC;
```

Mark: 4 marks — correct calculation (1), AS alias (1), GROUP BY (1), ORDER BY DESC (1)

Q25 [Medium] *Average risk_weight per Danger_type where avg > 1.5 as AvgRisk.*

```
SELECT Danger_type, AVG(risk_weight) AS AvgRisk
FROM Danger
GROUP BY Danger_type
HAVING AVG(risk_weight) > 1.5;
```

Mark: 4 marks — AVG (1), GROUP BY (1), HAVING (1), correct threshold (1)

Q26 [Medium] *DangerCode using LEFT 3 of Danger_type + RIGHT 2 of location_zone.*

```
SELECT Danger_type, location_zone,
       LEFT(Danger_type, 3) & RIGHT(location_zone, 2) AS DangerCode
FROM Danger;
```

Mark: 4 marks — LEFT correct (1), RIGHT correct (1), & concatenation (1), alias (1)

Q27 [Medium] *Danger_description character count as DescLength.*

```
SELECT Danger_type, Danger_description,
       LEN(Danger_description) AS DescLength
FROM Danger;
```

Mark: 3 marks — correct fields (1), LEN function (1), alias (1)

Q28 [Medium] *AdjustedCost = action_cost x people_involved, sorted descending.*

```
SELECT action_name,  
       (action_cost * people_involved) AS AdjustedCost  
FROM Actions  
ORDER BY AdjustedCost DESC;
```

Mark: 4 marks — correct fields selected (1), correct calculation (1), alias (1), ORDER BY DESC (1)

Q29 [Medium] *Count of recommended actions per Danger_type, sorted descending.*

```
SELECT Danger_type, COUNT(action_id) AS RecommendedCount  
FROM Danger, Decision_link  
WHERE Danger.Danger_id = Decision_link.Danger_id  
AND recommended = True  
GROUP BY Danger_type  
ORDER BY RecommendedCount DESC;
```

Mark: 5 marks — tables joined (1), WHERE join condition (1), recommended filter (1), GROUP BY (1), ORDER BY DESC (1)

Q30 [Medium] *Distinct location zones per Danger_type where UniqueZones > 1.*

```
SELECT Danger_type, COUNT(DISTINCT location_zone) AS UniqueZones  
FROM Danger  
GROUP BY Danger_type  
HAVING COUNT(DISTINCT location_zone) > 1;
```

Mark: 4 marks — DISTINCT used (1), GROUP BY (1), HAVING (1), correct condition (1)

MEMO: PART 3 — HARD (Q31–Q43)

Q31 [Hard] *action_name and Danger_type from all three tables.*

```
SELECT action_name, Danger_type
FROM Actions, Danger, Decision_link
WHERE Actions.action_id = Decision_link.action_id
AND Danger.Danger_id = Decision_link.Danger_id;
```

Mark: 4 marks — correct fields (1), all 3 tables (1), first join condition (1), second join condition (1)

Q32 [Hard] *Recommended actions with priority_score sorted descending.*

```
SELECT action_name, Danger_type, priority_score
FROM Actions, Danger, Decision_link
WHERE Actions.action_id = Decision_link.action_id
AND Danger.Danger_id = Decision_link.Danger_id
AND recommended = True
ORDER BY priority_score DESC;
```

Mark: 5 marks — correct fields (1), join conditions (1), recommended filter (1), ORDER BY DESC (1), all tables listed (1)

Q33 [Hard] *Linked records in School Zone with action and danger details.*

```
SELECT action_name, Danger_type, Danger_description, location_zone
FROM Actions, Danger, Decision_link
WHERE Actions.action_id = Decision_link.action_id
AND Danger.Danger_id = Decision_link.Danger_id
AND location_zone = 'School Zone';
```

Mark: 5 marks — correct 4 fields (1), all tables (1), both join conditions (1), location filter (1), exact string match (1)

Q34 [Hard] *Non-recommended actions with notes.*

```
SELECT action_name, Danger_type, notes
FROM Actions, Danger, Decision_link
WHERE Actions.action_id = Decision_link.action_id
AND Danger.Danger_id = Decision_link.Danger_id
AND recommended = False;
```

Mark: 4 marks — correct fields (1), all tables (1), join conditions (1), False filter (1)

Q35 [Hard] *Linked records where priority_score BETWEEN 8 AND 10.*

```
SELECT action_name, Danger_type, priority_score
FROM Actions, Danger, Decision_link
WHERE Actions.action_id = Decision_link.action_id
AND Danger.Danger_id = Decision_link.Danger_id
AND priority_score BETWEEN 8 AND 10;
```

Mark: 5 marks — correct fields (1), all tables (1), both join conditions (1), BETWEEN used (1), correct range (1)

Q36 [Hard] *YEAR, MONTH, DAY extracted from incident_date.*

```
SELECT Danger_type, Danger_description,  
       YEAR(incident_date) AS IncidentYear,  
       MONTH(incident_date) AS IncidentMonth,  
       DAY(incident_date) AS IncidentDay  
FROM Danger, Decision_link  
WHERE Danger.Danger_id = Decision_link.Danger_id;
```

Mark: 5 marks — YEAR (1), MONTH (1), DAY (1), correct aliases (1), correct join (1)

Q37 [Hard] *Linked incidents from current year using YEAR(DATE()).*

```
SELECT Danger_type, Danger_description, incident_date, location_zone  
FROM Danger, Decision_link  
WHERE Danger.Danger_id = Decision_link.Danger_id  
AND YEAR(incident_date) = YEAR(DATE());
```

Mark: 4 marks — correct fields (1), correct join (1), YEAR() function (1), DATE() for current year (1)

Q38 [Hard] *Linked records: risk_weight=3 AND recommended=True.*

```
SELECT action_name, Danger_type  
FROM Actions, Danger, Decision_link  
WHERE Actions.action_id = Decision_link.action_id  
AND Danger.Danger_id = Decision_link.Danger_id  
AND risk_weight = 3  
AND recommended = True;
```

Mark: 5 marks — fields (1), all tables (1), join conditions (1), risk_weight filter (1), recommended filter (1)

Q39 [Hard] *Average priority_score per Danger_type where AvgPriority > 6.*

```
SELECT Danger_type, AVG(priority_score) AS AvgPriority  
FROM Danger, Decision_link  
WHERE Danger.Danger_id = Decision_link.Danger_id  
GROUP BY Danger_type  
HAVING AVG(priority_score) > 6;
```

Mark: 5 marks — AVG (1), join (1), GROUP BY (1), HAVING (1), correct threshold (1)

Q40 [Hard] *Actions linked to more than 3 Dangers, sorted descending.*

```
SELECT action_name, COUNT(Danger_id) AS NumDangersLinked  
FROM Actions, Decision_link
```

```
WHERE Actions.action_id = Decision_link.action_id
GROUP BY action_name
HAVING COUNT(Danger_id) > 3
ORDER BY NumDangersLinked DESC;
```

Mark: 5 marks — COUNT (1), join (1), GROUP BY (1), HAVING > 3 (1), ORDER BY DESC (1)

Q41 [Hard] *Recommended Pedestrian dangers sorted by people_count descending.*

```
SELECT action_name, Danger_type, people_count
FROM Actions, Danger, Decision_link
WHERE Actions.action_id = Decision_link.action_id
AND Danger.Danger_id = Decision_link.Danger_id
AND Danger_type = 'Pedestrian'
AND recommended = True
ORDER BY people_count DESC;
```

Mark: 5 marks — all 3 tables (1), join conditions (1), Pedestrian filter (1), recommended (1), ORDER BY DESC (1)

Q42 [Hard] *SimCode = first 2 letters of action_name + action_id.*

```
SELECT action_name,
       LEFT(action_name, 2) & action_id AS SimCode
FROM Actions;
```

Mark: 3 marks — LEFT(2) (1), concatenation with & (1), action_id included (1)

Q43 [Hard] *Linked records where notes contains 'risk', sorted by priority_score DESC.*

```
SELECT action_name, Danger_type, priority_score
FROM Actions, Danger, Decision_link
WHERE Actions.action_id = Decision_link.action_id
AND Danger.Danger_id = Decision_link.Danger_id
AND notes LIKE '*risk*'
ORDER BY priority_score DESC;
```

Mark: 5 marks — fields (1), all tables (1), join conditions (1), LIKE pattern (1), ORDER BY DESC (1)

MEMO: PART 4 — VERY HARD (Q44–Q50)

Q44 [Very Hard] *Dangers with highest risk_weight using subquery.*

```
SELECT Danger_type, Danger_description, risk_weight
FROM Danger
WHERE risk_weight = (SELECT MAX(risk_weight) FROM Danger);
```

Mark: 4 marks — outer SELECT fields (1), subquery using MAX (1), correct subquery table (1), WHERE = subquery (1)

Q45 [Very Hard] *Actions never recommended using NOT IN subquery.*

```
SELECT action_name
FROM Actions
WHERE action_id NOT IN
      (SELECT action_id FROM Decision_link WHERE recommended = True);
```

Mark: 4 marks — NOT IN (1), correct outer table (1), subquery returns action_ids (1), recommended=True filter in subquery (1)

Q46 [Very Hard] *Recommended actions for highest total risk Danger using subquery.*

```
SELECT action_name, Danger_type, priority_score
FROM Actions, Danger, Decision_link
WHERE Actions.action_id = Decision_link.action_id
AND Danger.Danger_id = Decision_link.Danger_id
AND recommended = True
AND (people_count * risk_weight) =
      (SELECT MAX(people_count * risk_weight) FROM Danger)
ORDER BY priority_score DESC;
```

Mark: 6 marks — all tables joined (1), recommended filter (1), calculated field in WHERE (1), subquery with MAX (1), correct calculation in subquery (1), ORDER BY (1)

Q47 [Very Hard] *UPDATE: increase priority_score by 2 for recommended + high risk.*

```
UPDATE Decision_link
SET priority_score = priority_score + 2
WHERE recommended = True
AND Danger_id IN
      (SELECT Danger_id FROM Danger WHERE risk_weight = 3);
```

Mark: 5 marks — UPDATE correct table (1), SET correct expression (1), recommended=True (1), Danger_id IN subquery (1), subquery correct (1)

Q48 [Very Hard] *INSERT: new Danger record for speeding truck on highway.*

```
INSERT INTO Danger (Danger_type, Danger_description, people_count, risk_weight,
location_zone)
```

```
VALUES ('Vehicle', 'Speeding truck', 3, 3, 'Highway');
```

Mark: 4 marks — correct table (1), all 5 fields listed (1), VALUES keyword (1), all 5 values in correct order and type (1)

Q49 [Very Hard] *DELETE: Decision_link where priority_score < 3 AND recommended = False.*

```
DELETE FROM Decision_link
WHERE priority_score < 3
AND recommended = False;
```

Mark: 4 marks — DELETE FROM correct table (1), priority_score < 3 (1), AND (1), recommended = False (1)

Q50 [Very Hard] *Average risk_weight of Dangers with more than 2 recommended actions using subquery.*

```
SELECT AVG(risk_weight) AS AvgRiskOfPopularDangers
FROM Danger
WHERE Danger_id IN (
    SELECT Danger_id
    FROM Decision_link
    WHERE recommended = True
    GROUP BY Danger_id
    HAVING COUNT(action_id) > 2
);
```

Mark: 6 marks — AVG outer query (1), correct alias (1), Danger_id IN subquery (1), GROUP BY in subquery (1), HAVING COUNT > 2 (1), recommended=True filter (1)

Mark Allocation Summary

Section	Questions	Total Marks	Difficulty
Part 1	Q1–Q15	35	Easy (avg 2.3 each)
Part 2	Q16–Q30	53	Medium (avg 3.5 each)
Part 3	Q31–Q43	62	Hard (avg 4.8 each)
Part 4	Q44–Q50	33	Very Hard (avg 4.7 each)
TOTAL	50 questions	183 marks	—

Good luck! 🚀