Haris Ahmed
23K-6005
BSAI-5A

SYSTEMS

DATABASE ASSIGNMENT 2 **Date:**

QUESTION 1:

Expression:

1. π Country_Name (

~~σ Severity_Level = 'Critical' (THREAT)~~ σ Date_Reported $\geq$ '2025-01-01' $\wedge$ Date_Reported $\leq$ '2025-12-31' (

σ Severity_Level = 'Critical' (THREAT) $\bowtie$ THREAT.Threat_ID = INCIDENT.Threat_ID (INCIDENT)

$\bowtie$ INCIDENT.Country_ID = COUNTRY.Country_ID COUNTRY))

Explanation:

- σ Severity_Level = 'Critical' (THREAT): Selects only threats where the severity level is marked as critical from the THREAT Relation.
- $\bowtie$ THREAT.Threat_ID = INCIDENT.Threat_ID INCIDENT: Joins the selected critical threats with incidents that reference those threats via Threat_ID.
- $\bowtie$ INCIDENT, Country_ID = COUNTRY.Country_ID COUNTRY: Joins the resulting incidents with countries based on Country_ID to associate incidents with country details.
- σ Date_Reported $\geq$ '2025-01-01' $\wedge$ Date_Reported $\leq$ '2025-12-31': Filters the joined result to include only incidents reported within the year 2025.
- π Country_Name: Projects only the country names from the filtered result, listing unique countries.

2. π Team_Name (

(σ Resolution_Status = 'Resolved' (

(σ Threat_Name = 'Ransomware' (THREAT) $\bowtie$ THREAT.Threat_ID = INCIDENT.Threat_ID) INCIDENT)

$\bowtie$ INCIDENT.Incident_ID = ACTION_TAKEN.Incident_ID ACTION_TAKEN))

$\bowtie$ ACTION_TAKEN.Team_ID = ~~Response~~ RESPONSE_TEAM.Team_ID RESPONSE_TEAM)

Explanation:

- σ Threat_Name = 'Ransomware' (THREAT): Selects threat specifically named 'Ransomware' from the THREAT Relation
- $\bowtie$ THREAT.Threat_ID = INCIDENT.Threat_ID INCIDENT: Joins the ransomware threats with incidents caused by those threats via Threat_ID.
- $\bowtie$ INCIDENT, Incident_ID = ACTION_TAKEN.Incident_ID ACTION_TAKEN: Joins the ransomware incidents with actions taken on those incidents via Incident_ID.
- σ Resolution_Status = 'Resolved': Filters the joined actions to include only those with a resolved status.

Date: _____

- ⋈ ACTION_TAKEN.Team_ID = RESPONSE_TEAM.TeamID RESPONSE_TEAM : Joins the resolved actions with response teams based on Team_ID to get team details
- π Team_Name : Projects only the names of the response teams from the final joined result.

3. π Country_Name (COUNTRY) − π Country_Name (
COUNTRY ⋈ COUNTRY.Country_ID = RESPONSE_TEAM.Country_ID RESPONSE_TEAM)

Explanations:

- COUNTRY ⋈ COUNTRY.Country_ID = RESPONSE_TEAM.Country_ID RESPONSE TEAM : Joins countries with response teams based on COUNTRY_ID to find countries that have at least one team.
- π Country_Name : Projects the names of countries that have response teams from the joined result.
- π Country_Name (COUNTRY) : Projects the names of all countries from the COUNTRY relation.
- (Set difference) : Subtracts the countries with teams from all countries, resulting in countries without any teams.

4. π Threat_Name (
(γ Threat_ID, COUNT (DISTINCT Country_ID) → Country_Count (INCIDENT) σ Country_Count > 1)
⋈ Threat_ID = THREAT.Threat_ID THREAT)

Explanation:

- γ Threat_ID, COUNT (DISTINCT Country_ID) → Country_Count (INCIDENT) : Groups incidents by Threat_ID and counts the distinct countries reporting each threat, renaming the count to Country_Count
- σ Country_Count > 1 : Selects only the groups where the distinct country count is greater than 1.
- ⋈ Threat_ID = THREAT.Threat_ID THREAT : Joins the filtered threat ID's with the THREAT relation to get threat details.
- π Threat_Name : Projects only the names of threats that meet the criteria.

5. π Incident_ID, Country_Name, Impact_Score (
σ Impact_Score > 80 (
(σ Category = 'AI Attack' (THREAT)
⋈ THREAT.Threat_ID = INCIDENT.Threat_ID INCIDENT) ⋈ INCIDENT.Country_ID = COUNTRY.Country_ID
COUNTRY))

Date: _____

Explanation:

- σ Category = 'AI Attack' (THREAT): Selects threats in the 'AI Attack' category from the THREAT relation.
- $\bowtie$ THREAT.Threat_ID = INCIDENT.Threat_ID INCIDENT: Joins the AI Attack threats with incidents referencing those threats via Threat_ID.
- $\bowtie$ INCIDENT.County_ID = COUNTRY.County_ID COUNTRY: Joins the incidents with countries based on County_ID to include county names.
- σ Impact_Score > 80: Filters the joined result to include only incidents with an impact score greater than 80.
- π Incident_ID, County_Name, Impact_Score: Projects the incident ID, county name, and impact score from the filtered result.

QUESTION 2:

(i) This produces a relation containing the names of menu items that are not available (isAvailable = 0) and have a price greater than 500.

2. This produces a relation containing the names of menu items that are either in the 'Beverage' category with a price greater than 300 or in the 'Dessert' category (regardless of price)

3. This produces a relation containing the names and prices of menu items that are in the 'Snack' category and are available (isAvailable = 1)

4. This produces a relation containing the names of menu items that are in the 'Snack' category and are available (isAvailable = 1), along with their prices divided by 100.

5. This produces a relation containing the supplier IDs of suppliers that have supplied the menu items named 'Cappuccino' (assuming the redundant menu Item join in the RA is a notation error and it's effectively Supplies joined with filtered MenuItems on ItemID).

Date: _____

6. This produces a relation containing the names of employees and the names of the cafes they work at, where the employees works at the cafe (joined on cafeID) and the employee's address matches the cafe's city (assuming address can be directly compared to city, eg. if address is the city name).

7. This produces a relation containing the employee IDs of employees whose salary is not equal to 50,000.

8. This produces a relation containing the supplier IDs that are common to suppliers located in 'Karachi' and suppliers located in 'Lahore' (i.e suppliers listed with both cities, which would typically be an empty relation if each supplier has a unique city).

9. This produces a relation containing the names of menu items that appear in the menuItems table but do not appear in any supplies (i.e menu items that have never been supplied, using set difference after joining menuItem and Supplies on itemID).