ANALYSIS OF GLOBAL TERRORISM DATABASE

COP5725 - Database Management System
Project Deliverable III
Database Schema Construction and Oracle Snapshots

NUMBER: 1

NAME /Email: Daye Kim (dayekim@ufl.edu)

PROFESSOR: Dr. Markus Schneider

Index

Introduction

Data Background

My data set is the global terrorist dataset found in kaggle. My dataset has over 250K tuples from the multiple entities recorded for 40 years from 1970 to 2017. This data set is maintained by the National Consortium for the study of Terrorism and Responses to Terrorism (START), headquartered at the University of Maryland. The data set contains about 135 different parameters for each individual event and I split into five entities with identifiable key and foreign keys struct the relationships among relevant entities. Each entity has plentiful and detailed information in distinct attributes and this data set contains information on time, weapons, property damage, location, and targets by terrorists.

Application Purpose

My application is aimed to improve public safety, and provide useful information and tools for public as well as professional institutions to find the attack type, attack geographical distribution, weapon types, powerful terrorist organization trends using the recent 47 years global terrorist attack data.
Through the analysis of fatality, location, time information of past terror attacks, the user can find different trends using the existing data and see the roughly possibility of a terror attack, and estimate the danger and damage it may cause by the user's own computation and estimation by using my trend data. Therefore, the professional user can take actions to prevent potential terror attacks occurring and the normal user can try to avoid the specific area containing potential risks and then minimize the damage.

What I have

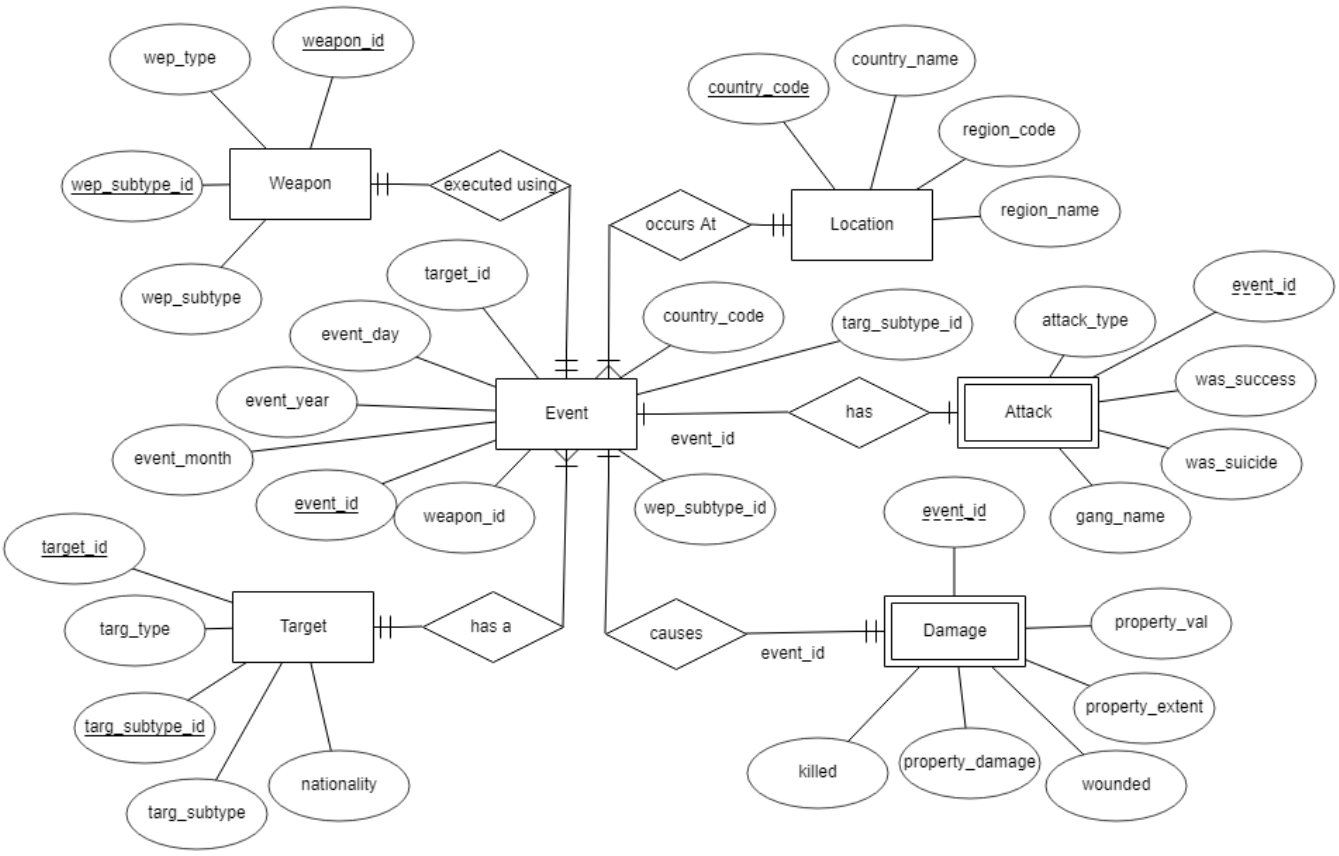
I have the dataset, trend queries, entities and their attributes, and the relation between each entity. The E-R diagram is established and user interface design is done.

To do in this version

I need to transform the E-R diagram into a relational data schema. The approach I used and the rule I followed is introduced in the Relational Database Schema section in details.

Creating the relation data schema using Oracle SQL. The SQL data definition language will be used to create empty tables as well as the attributes and data type. The entire architecture of my database will be created in this version. More details and specific SQL statements are introduced in the Oracle queries and results section.

Entity-Relationship Diagram



S. No.	Relationship	E1	E2	Cardinalities
1	EXECUTED USING	Event	Weapon	1:m. An event is executed using a weapon. A weapon can be used in multiple events.
2	HAS A	Event	Target	1:m. An event can have one target. A target can be attacked in multiple events.
3	CAUSES	Event	Damage (Weak)	1:1. An event can have one damage. A damage is attached to an event.
4	OCCURS AT	Event	Location	1:m. An event occurs at one location. A location can be attacked in multiple events.
5.	HAS	Event	Attack (Weak)	1:1. An event is attached to one attack detail. One attack detail is attached to one attack.

Transforming E-R diagram into Relational Database Schema

General Representation

E ($\underline{A_1}$: $\underline{D_1}$,
A_2 (foreign key): D_2
….
….
….
A_n:D_n)

Here,
E = Entity, $A_{1...n}$ = Attributes, $D_{1...n}$ = Data Types.
An underline under an attribute means that it is the primary key for this entity.
An attribute followed by (foreign key) means that it is a foreign key attribute that is referencing a key from a different table.

Transformation Rules Used

1. For a relationship with 1:m cardinality, the primary key of the relation with cardinality = 1, becomes the foreign key in the relation with cardinality = m.

2. Every weak entity set borrows the primary key of the associated strong entity set since it doesn't have a primary key of its own.

3. For all other situations, determine the data type of each attribute and set the attribute and data type along with their strong entity.

    a. Integer: If the value of an attribute is numeric value and between -2147483648 to 2147483647, then the data type of it is set to 'INTEGER'.

    b. Varchar: If the value of an attribute contains a different length of string, then the data type of it is set to 'VARCHAR'

Relational Database Schema

1. WEAPON (<u>weapon_id</u>: *INTEGER*,
   wep_type: *VARCHAR*,
   <u>wep_subtype_id</u>: *INTEGER*,
   wep_subtype: *VARCHAR*)

```
CREATE TABLE WEAPON(
weapon_id INTEGER,
wep_type VARCHAR(25),
wep_subtype_id INTEGER,
wep_subtype VARCHAR(40),
PRIMARY KEY (weapon_id,wep_subtype_id)
);
```

WEAPON

Columns | Data | Model | Constraints | Grants | Statistics | Triggers | Flashback | Dependencies | Details | Partitions | Indexes | SQL

Actions...

	COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
1	WEAPON_ID	NUMBER(38,0)	No	(null)	1	(null)
2	WEP_TYPE	VARCHAR2(25 BYTE)	Yes	(null)	2	(null)
3	WEP_SUBTYPE_ID	NUMBER(38,0)	No	(null)	3	(null)
4	WEP_SUBTYPE	VARCHAR2(40 BYTE)	Yes	(null)	4	(null)

Explanation:

- <u>WEAPON_ID</u>: the unique weapon identity that each weapon type has. This is a specific code that varies from integer -9 to 12.
- TYPE: a string referring to the name of a weapon type.
- <u>WEP_SUBTYPE_ID</u>: the unique sub weapon identity that each sub weapon type has. This is a specific code that varies from integer -9 to 31.
- WEP_SUBTYPE: a string referring to the name of a sub weapon type.
- This table has two primary keys, WEAPON_ID and WEP_SUBTYPE_ID, due to their combined uniqueness.

2. TARGET (<u>target_id</u>: *INTEGER*,
   targ_type; *VARCHAR*,
   <u>targ_subtype_id</u>: *INTEGER*,
   targ_subtype: *VARCHAR*,
   nationality: *VARCHAR*)

```
CREATE TABLE TARGET(
target_id INTEGER,
targ_type VARCHAR(40),
targ_subtype_id INTEGER,
targ_subtype VARCHAR(60),
nationality VARCHAR(40),
PRIMARY KEY(target_id,targ_subtype_id)
);
```

TARGET

Columns | Data | Model | Constraints | Grants | Statistics | Triggers | Flashback | Dependencies | Details | Partitions | Indexes | SQL

Actions...

	COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
1	TARGET_ID	NUMBER(38,0)	No	(null)	1	(null)
2	TARG_TYPE	VARCHAR2(40 BYTE)	Yes	(null)	2	(null)
3	TARG_SUBTYPE_ID	NUMBER(38,0)	No	(null)	3	(null)
4	TARG_SUBTYPE	VARCHAR2(60 BYTE)	Yes	(null)	4	(null)
5	NATIONALITY	VARCHAR2(40 BYTE)	Yes	(null)	5	(null)

Explanation:

- TARGET_ID: the unique target identity that each target type has. This is a specific code that varies from integer -9 to 12.
- TYPE: a string referring to the name of a target type.
- TARG_SUBTYPE_ID: the unique target identity that each target subtype has. This is a specific code that varies from integer -9 to 113.
- TARG_SUBTYPE: a string referring to the name of a target subtype.
- NATIONALITY: a string referring to the name of countries in which the target is located.
- This table has two primary keys, TARGET_ID and TARG_SUBTYPE_ID, due to their combined uniqueness.

3. EVENT (event_id: *INTEGER*,
   event_day: *INTEGER*,
   event_month: *INTEGER*,
   event_year: *INTEGER*,
   weapon_id (foreign key): *INTEGER*,
   wep_subtype_id(foreign key): INTEGER,
   target_id (foreign key):*INTEGER*,
   targ_subtype_id (foreign key):*INTEGER*,
   country_code (foreign key): *INTEGER*)

```
CREATE TABLE EVENT (event_id INTEGER,
                    event_day INTEGER,
                    event_month INTEGER,
                    event_year INTEGER,
                    weapon_id INTEGER,
                    wep_subtype_id INTEGER,
                    target_id INTEGER,
                    targ_subtype_id INTEGER,
                    country_code INTEGER,
                    PRIMARY KEY (event_id),
                    FOREIGN KEY(weapon_id,wep_subtype_id)REFERENCES WEAPON(weapon_id,wep_subtype_id),
                    FOREIGN KEY(target_id,targ_subtype_id)REFERENCES TARGET(target_id,targ_subtype_id),
                    FOREIGN KEY(country_code)REFERENCES LOCATION(country_code)
                   );
```

EVENT

Columns | Data | Model | Constraints | Grants | Statistics | Triggers | Flashback | Dependencies | Details | Partitions | Indexes | SQL

Actions...

	COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
1	EVENT_ID	NUMBER(38,0)	No	(null)	1	(null)
2	EVENT_DAY	NUMBER(38,0)	Yes	(null)	2	(null)
3	EVENT_MONTH	NUMBER(38,0)	Yes	(null)	3	(null)
4	EVENT_YEAR	NUMBER(38,0)	Yes	(null)	4	(null)
5	WEAPON_ID	NUMBER(38,0)	Yes	(null)	5	(null)
6	WEP_SUBTYPE_ID	NUMBER(38,0)	Yes	(null)	6	(null)
7	TARGET_ID	NUMBER(38,0)	Yes	(null)	7	(null)
8	TARG_SUBTYPE_ID	NUMBER(38,0)	Yes	(null)	8	(null)
9	COUNTRY_CODE	NUMBER(38,0)	Yes	(null)	9	(null)

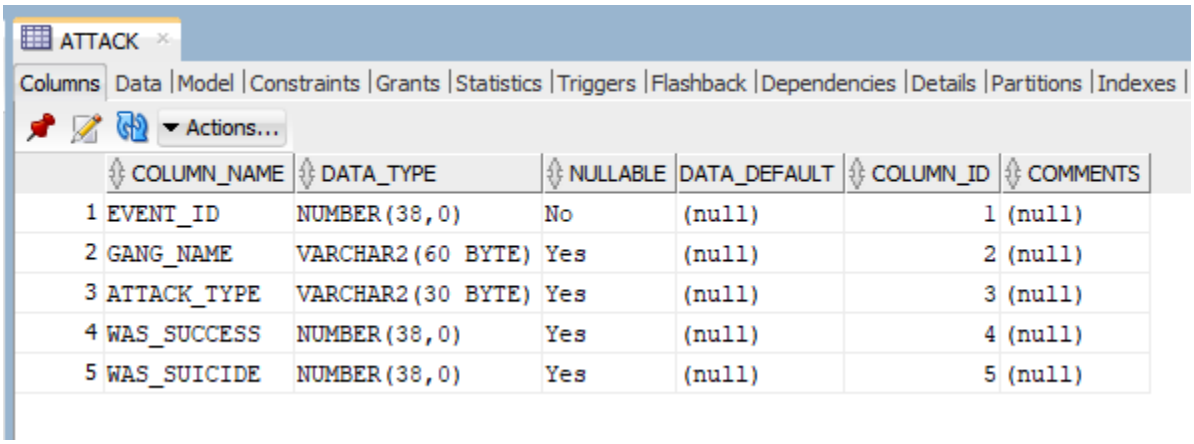
Explanation:

- EVENT_ID: unique 12-digit event identity in which the first 8-digit is date recorded in “yyyymmdd” format and the last 4 digitals is the sequence the different events took place for that day.
- EVENT_DAY: 2-digit integer from 1 to 31 that specifies the day of the month that the event took place.
- EVENT_MONTH: 2-digit integer from 1 to 12 that specifies the month the event took place.
- EVENT_YEAR: 4-digit identity integer from 1970 to 2017 that specifies the year the event took place.
- WEAPON_ID: This is a foreign key referencing WEAPON_ID in the WEAPON entity.
- WEP_SUBTYPE_ID: This is a foreign key referencing WEP_SUBTYPE_ID in the WEAPON entity.
- TARGET_ID: This is a foreign key referencing TARGET_ID in the TARGET entity.

- TARGET_SUBTYPE_ID: This is a foreign key referencing TARG_SUBTYPE_ID in the TARGET entity.
- COUNTRY_CODE: This is a foreign key referencing COUNTRY_CODE in the LOCATION entity.
- This table has one primary key, EVENT_ID, due to its uniqueness.

4. ATTACK (<u>event_id (foreign key):</u> *INTEGER*,
   gang_name: *VARCHAR*,
   attack_type: *VARCHAR*,
   was_success: *INTEGER*,
   was_suicide: *INTEGER*)

```
CREATE TABLE ATTACK (event_id INTEGER,
    gang_name VARCHAR(60),
    attack_type VARCHAR(30),
    was_success INTEGER,
    was_suicide INTEGER,
    PRIMARY KEY(event_id),
    FOREIGN KEY (event_id) REFERENCES EVENT(event_id));
```


ATTACK

Columns | Data | Model | Constraints | Grants | Statistics | Triggers | Flashback | Dependencies | Details | Partitions | Indexes

Actions...

	COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
1	EVENT_ID	NUMBER(38,0)	No	(null)	1	(null)
2	GANG_NAME	VARCHAR2(60 BYTE)	Yes	(null)	2	(null)
3	ATTACK_TYPE	VARCHAR2(30 BYTE)	Yes	(null)	3	(null)
4	WAS_SUCCESS	NUMBER(38,0)	Yes	(null)	4	(null)
5	WAS_SUICIDE	NUMBER(38,0)	Yes	(null)	5	(null)

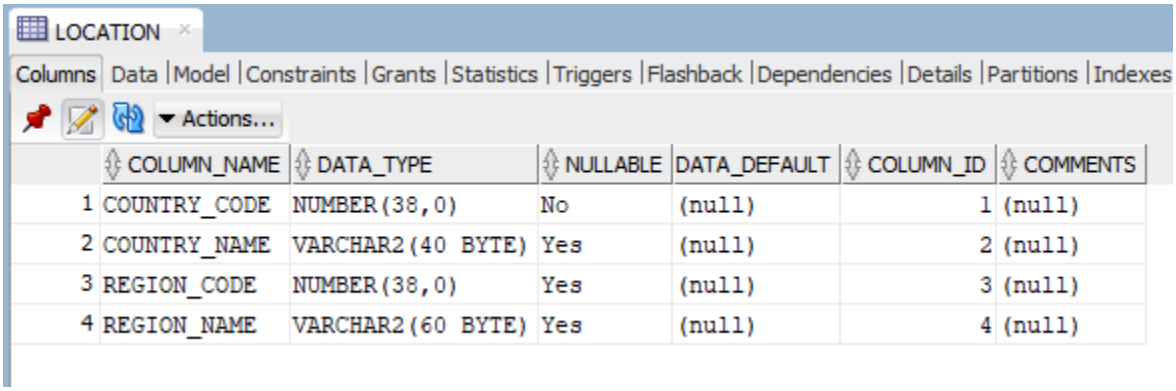
Explanation:

- <u>EVENT_ID</u>: This is a foreign key referencing EVENT_ID from the EVENT entity.
- GANG_NAME: a string referring to the name of a terrorist organization for a specific event.
- ATTACK_NAME: a string referring to the type of attack for an event; includes bombing/explosion, facility attack etc.
- WAS_SUCCESS: an integer value where 0 is false and 1 is true. False means the attack failed and true means the attack was a success.
- WAS_SUICIDE: an integer value where 0 is false and 1 is true. False means the attack was a suicide attack and true means the attack was not.

- ATTACK is a weak entity set that references EVENT_ID from the EVENT table as its primary key.

5. LOCATION (<u>country_code</u>: *INTEGER*,
   country_name: *VARCHAR*,
   region_code: *INTEGER*,
   region_name: *VARCHAR*)

```
CREATE TABLE LOCATION (country_code INTEGER,
                       country_name VARCHAR(40),
                       region_code INTEGER,
                       region_name VARCHAR(60),
                       PRIMARY KEY(country_code));
```



LOCATION

Columns | Data | Model | Constraints | Grants | Statistics | Triggers | Flashback | Dependencies | Details | Partitions | Indexes

Actions...

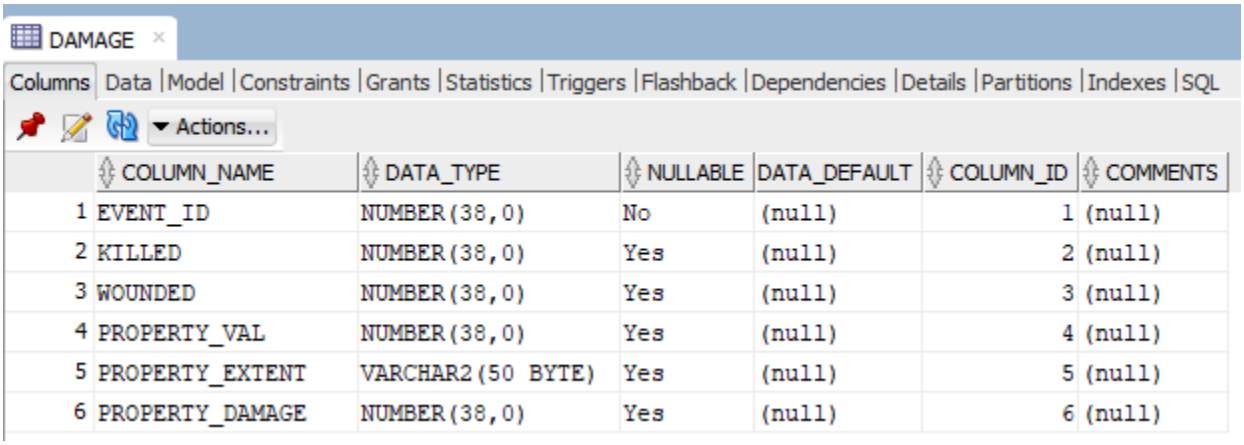
	COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
1	COUNTRY_CODE	NUMBER(38,0)	No	(null)	1	(null)
2	COUNTRY_NAME	VARCHAR2(40 BYTE)	Yes	(null)	2	(null)
3	REGION_CODE	NUMBER(38,0)	Yes	(null)	3	(null)
4	REGION_NAME	VARCHAR2(60 BYTE)	Yes	(null)	4	(null)

Explanation:

- <u>COUNTRY_CODE</u>: the country code where each country has a unique integer value from 4 to 1004.
- COUNTRY_NAME: a string referring to the name of a country.
- REGION_CODE: the region code where we list 12 regions that have a unique integer value from 1 to 12.
- REGION_NAME: a string referring to the name of a specific region.
- This table has one primary key, COUNTRY_CODE, due to its uniqueness.

6. DAMAGE (<u>event_id (foreign key)</u>: *INTEGER*,
   killed: *INTEGER*,
   wounded: *INTEGER*,
   property_val: *INTEGER*,
   property_extent: *VARCHAR*,
   property_damage: *INTEGER*)

```
CREATE TABLE DAMAGE (event_id INTEGER,
                     killed INTEGER,
                     wounded INTEGER,
                     property_val INTEGER,
                     property_extent VARCHAR(50),
                     property_damage INTEGER,
                     PRIMARY KEY (event_id),
                     FOREIGN KEY (event_id)REFERENCES EVENT(event_id));
```



DAMAGE

Columns | Data | Model | Constraints | Grants | Statistics | Triggers | Flashback | Dependencies | Details | Partitions | Indexes | SQL

Actions...

	COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
1	EVENT_ID	NUMBER(38,0)	No	(null)	1	(null)
2	KILLED	NUMBER(38,0)	Yes	(null)	2	(null)
3	WOUNDED	NUMBER(38,0)	Yes	(null)	3	(null)
4	PROPERTY_VAL	NUMBER(38,0)	Yes	(null)	4	(null)
5	PROPERTY_EXTENT	VARCHAR2(50 BYTE)	Yes	(null)	5	(null)
6	PROPERTY_DAMAGE	NUMBER(38,0)	Yes	(null)	6	(null)

Explanation:

- EVENT_ID: This is a foreign key referencing EVENT_ID in the EVENT entity.
- KILLED: an integer from -9 to 1570 indicating how many people were killed in a specific terrorist event.
- WOUNDED: an integer from -9 to 1570 indicating how many people were wounded in a specific terrorist event.
- PROPERTY_VAL: an integer from -9 to 1570 indicating the exact amount of property damage done in USD.
- PROPERTY_EXTENT: a string referring to rough damage estimation e.g. ‘Minor (likely < $1 million)’.
- PROPERTY_DAMAGE: an integer contains -9, 0 and 1 indicating if there is property damage in an event. 0 is false, 1 is true, and -9 is unknown.
- DAMAGE is a weak entity set that references EVENT_ID from the EVENT table as its primary key.

Note: *Integer -9 in this database indicates unknown for any entities or attributes.*