**Goal #1**

**Keep track of who is calling and who is being called. This includes the following parameters:**

1. Calling Party Identity: This refers to the phone number or other identifier of the person who made the call.

2. Called Party Identity: This refers to the phone number or other identifier of the person who received the call.

3. Date and Time Called: This is the date and time that the call was initiated.

4. Date and Time Call End: This is the date and time that the call ended.

5. Duration In minutes: This is the length of the call in minutes.

6. Duration In Second: This is the length of the call in seconds.

7. Duration in hrs: This is the length of the call in hours.

8. Location call made: This is the physical location (e.g. city, state, or country) where the call was initiated.

9. Locals call to: This is the physical location (e.g. city, state, or country) where the call was received.

10. Call type (audio, Video, SMS): This is the type of call, such as voice (audio), video, or SMS (text message).

**Tabel Fields : CDR**

1. Index auto increment
2. Calling Party Identity
3. Called Party Identity
4. Data and time called
5. Data and time call end
6. Duration In minuts
7. Duration In Second
8. Duration in hrs
9. Location call made
10. Local called to
11. Call type (audio , Video,Sms)

**The CDR table stores the details of each call made, including the call\_type\_id column. The call\_type\_id column is a foreign key that references the id column in the call\_type\_options table. The call\_type\_options table contains a list of all the available call types (Audio, Video, SMS) with their corresponding id and name value**

**Create Table**

**SQL Server Format :**

CREATE TABLE call\_type\_options (

call\_type\_id INT IDENTITY(1,1) PRIMARY KEY,

call\_type\_name VARCHAR(20) UNIQUE

);

INSERT INTO call\_type\_options (call\_type\_name) VALUES

('audio'),

('video'),

('SMS');

CREATE TABLE CDR (

record\_number INT IDENTITY(1,1) PRIMARY KEY,

calling\_party\_identity VARCHAR(20) UNIQUE,

called\_party\_identity VARCHAR(20) UNIQUE,

date\_time\_called DATETIME,

date\_time\_call\_end DATETIME,

duration\_minutes INT,

duration\_seconds INT,

duration\_hours INT,

location\_call\_made VARCHAR(50),

location\_call\_to VARCHAR(50),

call\_type\_id INT,

FOREIGN KEY (call\_type\_id) REFERENCES call\_type\_options (call\_type\_id)

);

CREATE INDEX ix\_calling\_party\_identity ON CDR (calling\_party\_identity);

CREATE INDEX ix\_called\_party\_identity ON CDR (called\_party\_identity);

**SQL Lite Format:**

CREATE TABLE call\_type\_options (

call\_type\_id INTEGER PRIMARY KEY AUTOINCREMENT,

call\_type\_name TEXT UNIQUE

);

INSERT INTO call\_type\_options (call\_type\_name) VALUES

('audio'),

('video'),

('SMS');

CREATE TABLE CDR (

record\_number INTEGER PRIMARY KEY AUTOINCREMENT,

calling\_party\_identity TEXT UNIQUE,

called\_party\_identity TEXT UNIQUE,

date\_time\_called TEXT,

date\_time\_call\_end TEXT,

duration\_minutes INTEGER,

duration\_seconds INTEGER,

duration\_hours INTEGER,

location\_call\_made TEXT,

location\_call\_to TEXT,

call\_type\_id INTEGER,

FOREIGN KEY (call\_type\_id) REFERENCES call\_type\_options (call\_type\_id)

);

CREATE INDEX ix\_calling\_party\_identity ON CDR (calling\_party\_identity);

CREATE INDEX ix\_called\_party\_identity ON CDR (called\_party\_identity);

**My SQL :**

CREATE TABLE call\_type\_options (

call\_type\_id INT PRIMARY KEY AUTO\_INCREMENT,

call\_type\_name VARCHAR(20) UNIQUE

);

INSERT INTO call\_type\_options (call\_type\_name) VALUES

('audio'),

('video'),

('SMS');

CREATE TABLE CDR (

record\_number INT PRIMARY KEY AUTO\_INCREMENT,

calling\_party\_identity VARCHAR(20) UNIQUE,

called\_party\_identity VARCHAR(20) UNIQUE,

date\_time\_called DATETIME,

date\_time\_call\_end DATETIME,

duration\_minutes INT,

duration\_seconds INT,

duration\_hours INT,

location\_call\_made VARCHAR(50),

location\_call\_to VARCHAR(50),

call\_type\_id INT,

FOREIGN KEY (call\_type\_id) REFERENCES call\_type\_options (call\_type\_id)

);

CREATE INDEX ix\_calling\_party\_identity ON CDR (calling\_party\_identity);

CREATE INDEX ix\_called\_party\_identity ON CDR (called\_party\_identity);

**PostgreSQL**

CREATE TABLE call\_type\_options (

call\_type\_id SERIAL PRIMARY KEY,

call\_type\_name VARCHAR(20) UNIQUE

);

INSERT INTO call\_type\_options (call\_type\_name) VALUES

('audio'),

('video'),

('SMS');

CREATE TABLE CDR (

record\_number SERIAL PRIMARY KEY,

calling\_party\_identity VARCHAR(20) UNIQUE,

called\_party\_identity VARCHAR(20) UNIQUE,

date\_time\_called TIMESTAMP,

date\_time\_call\_end TIMESTAMP,

duration\_minutes INTEGER,

duration\_seconds INTEGER,

duration\_hours INTEGER,

location\_call\_made VARCHAR(50),

location\_call\_to VARCHAR(50),

call\_type\_id INTEGER REFERENCES call\_type\_options (call\_type\_id)

);

CREATE INDEX ix\_calling\_party\_identity ON CDR (calling\_party\_identity);

CREATE INDEX ix\_called\_party\_identity ON CDR (called\_party\_identity);

**Gola # 2**

**Pricing for the call using a price table**

Price table:

1. Call Location Country
2. Call Location City
3. Cost per minute.
4. Round to
5. Call type Audio/Video/SMS

**The price table stores the cost details for each call based on the call\_type\_id, call\_location\_country, and call\_location\_city. The call\_type\_id column in the price table is also a foreign key that references the id column in the call\_type\_options table**

**Sql Server :**

CREATE TABLE price (

id INT PRIMARY KEY IDENTITY(1,1),

call\_location\_country VARCHAR(50),

call\_location\_city VARCHAR(50),

cost\_per\_minute DECIMAL(10,2),

round\_to INT,

call\_type\_id INT,

FOREIGN KEY (call\_type\_id) REFERENCES call\_type\_options (id)

);

**SQL Lite :**

CREATE TABLE price (

id INTEGER PRIMARY KEY AUTOINCREMENT,

call\_location\_country TEXT,

call\_location\_city TEXT,

cost\_per\_minute REAL,

round\_to INTEGER,

call\_type\_id INTEGER,

FOREIGN KEY (call\_type\_id) REFERENCES call\_type\_options (id)

);

**MYSQL:**

CREATE TABLE price (

id INT PRIMARY KEY AUTO\_INCREMENT,

call\_location\_country VARCHAR(50),

call\_location\_city VARCHAR(50),

cost\_per\_minute DECIMAL(10,2),

round\_to INT,

call\_type\_id INT,

FOREIGN KEY (call\_type\_id) REFERENCES call\_type\_options (id)

);

**Postgres SQL :**

CREATE TABLE price (

id SERIAL PRIMARY KEY,

call\_location\_country VARCHAR(50),

call\_location\_city VARCHAR(50),

cost\_per\_minute NUMERIC(10,2),

round\_to INTEGER,

call\_type\_id INTEGER REFERENCES call\_type\_options (id)

);

**Relationship:**

Here's an example SQL query that demonstrates the relationships between the CDR, call\_type\_options, and price tables:

**This query joins the CDR, price, and call\_type\_options tables to retrieve the details of a specific call from the CDR table, including the cost per minute and the round-to value from the price table, and the call type name from the call\_type\_options table. Note that you will need to replace <insert CDR ID here> with the actual ID of the call you want to retrieve.**

SELECT

cdr.id,

cdr.calling\_party\_identity,

cdr.called\_party\_identity,

cdr.date\_time\_called,

cdr.date\_time\_call\_end,

cdr.duration\_in\_minutes,

cdr.duration\_in\_seconds,

cdr.duration\_in\_hours,

cdr.location\_call\_made,

cdr.location\_call\_to,

call\_type\_options.call\_type\_name,

price.cost\_per\_minute,

price.round\_to

FROM

cdr

JOIN price ON

cdr.call\_type\_id = price.call\_type\_id

AND cdr.location\_call\_made = price.call\_location\_country

AND cdr.location\_call\_to = price.call\_location\_city

JOIN call\_type\_options ON

cdr.call\_type\_id = call\_type\_options.id

WHERE

cdr.id = <insert CDR ID here>;

**Goal # 3:**

**Generate a report :**

1. CDR : Call detail report provide information about call group by call location

* Calling Party Identity
* Called Party Identity
* Data and time called.
* Data and time call end
* Duration In minutes
* Duration In Second
* Duration in hrs.
* Location call made.
* Local called to
* Call type (Audio, Video, SMS)
  + Price paid.

**This query retrieves the details of all calls made between the start\_date and end\_date parameters, including the call type name, cost per minute, round-to value, and call cost. The call cost is calculated by multiplying the call duration in minutes by the cost per minute.**

You will need to replace <insert start date here> and <insert end date here> with the actual start and end dates for the report. Additionally, you can add other filters to the WHERE clause to further refine the report, such as filtering by specific locations, call types, or phone numbers.

SELECT

cdr.id,

cdr.calling\_party\_identity,

cdr.called\_party\_identity,

cdr.date\_time\_called,

cdr.date\_time\_call\_end,

cdr.duration\_in\_minutes,

cdr.duration\_in\_seconds,

cdr.duration\_in\_hours,

cdr.location\_call\_made,

cdr.location\_call\_to,

call\_type\_options.call\_type\_name,

price.cost\_per\_minute,

price.round\_to,

(cdr.duration\_in\_minutes \* price.cost\_per\_minute) AS call\_cost

FROM

cdr

JOIN price ON

cdr.call\_type\_id = price.call\_type\_id

AND cdr.location\_call\_made = price.call\_location\_country

AND cdr.location\_call\_to = price.call\_location\_city

JOIN call\_type\_options ON

cdr.call\_type\_id = call\_type\_options.id

WHERE

cdr.date\_time\_called >= <insert start date here>

AND cdr.date\_time\_called < <insert end date here>;

1. CSR : Call Summary report gives total cost per the call for group by location with time
   * Subtotal & Grand total cost by each customer

**This query retrieves the total duration and cost of all calls made from each location, broken down by call type. The WITH ROLLUP clause adds a subtotal row for each location and a grand total row for all locations.**

SELECT

cdr.location\_call\_made,

call\_type\_options.call\_type\_name,

SUM(cdr.duration\_in\_minutes) AS total\_duration\_in\_minutes,

SUM(cdr.duration\_in\_minutes \* price.cost\_per\_minute) AS total\_cost\_per\_location

FROM

cdr

JOIN price ON

cdr.call\_type\_id = price.call\_type\_id

AND cdr.location\_call\_made = price.call\_location\_country

AND cdr.location\_call\_to = price.call\_location\_city

JOIN call\_type\_options ON

cdr.call\_type\_id = call\_type\_options.id

GROUP BY

cdr.location\_call\_made,

call\_type\_options.call\_type\_name

WITH ROLLUP;

**Entity Relationship:**

Diagram, schematic

Description automatically generated

+--------------+