Lottery Sales in Texas

Business Analysis Template

Gabriela Cretu

June 2025

Contents

1 Business Description

1.1 Business Background

Lottery games, or luck-based games as we might call them, are part of a large and profitable industry around the world. People keep playing because of the strong desire to win big—sometimes just by scratching a ticket or picking a few lucky numbers. These games continue to be popular in many places.

In the United States, the lottery market is very large, and Texas stands out as one of the top states in terms of ticket sales. That's why this project focuses specifically on the Texas lottery. There are still many important questions to explore: Which games are the most or least profitable? What types of games do people in Texas prefer? Are instant-win scratch games more popular, or do players prefer waiting for draw results?

To find answers, it is important to understand how consumers behave, how well different games perform, and how retailers contribute to sales. This can be done by collecting and analyzing structured sales data from across Texas, which will help reveal useful insights and support better business decisions in this growing industry.

1.2 Problems Due to Poor Data Management

Poor data management can significantly hinder success in the lottery business. Without proper data, it's difficult to know which games are doing well and which ones are not. If you don't use tools that help you collect and analyze sales information, you won't be able to understand what players want or how retailers are performing.

In a competitive market like the Texas lottery, not having the right data can lead to missed opportunities, poor planning, and less profit. To stay competitive and make smart decisions, it's important to manage data properly and use it to create effective strategies.

1.3 Benefits from Implementing a Data Warehouse

Using a data warehouse can help solve the problems mentioned above. Implementing a data warehouse can answer important questions like:

- Which lottery games generate the most revenue?
- Which games have the widest range of sales across different retailers?
- Are there clear patterns in player preferences for instant-win versus draw-based games?

Further analysis of the data can also help to:

- Understand how sales vary by region or retailer type.
- Identify trends in customer behavior over time.
- Improve marketing strategies by targeting popular games.
- And many other useful insights.

By collecting and organizing data in one place, a data warehouse makes it easier to analyze sales, support better decision-making, and improve overall business performance.

1.4 Datasets Description

1.4.1 Scratch-Based Lottery

This dataset contains detailed transactional data for instant-win scratch tickets sold in Texas. Unlike the draw-based lottery, the assumption here is that a customer can purchase up to three tickets per transaction but can only win once per transaction because prizes are determined by ticket sellers. The prize value is represented by the `base_prize_scaled` attribute in the `DimPrizeRule` table.

Fact Table: ScratchTicketSalesFact

- **sale_id**(PK): unique identifier for the sales record
- **date_id_surrogate_id**(FK): foreign key to `DimDate`
- **game_scratch_surrogate_id**(FK): foreign key to `DimScratchGame`
- **retailer_license_number_surrogate_id**(FK): foreign key to `DimRetailer`
- **customer_id_surrogate_id**(FK): foreign key to `DimCustomer`
- **employee_id_surrogate_id**(FK): foreign key to `DimEmployee`
- **prize_rule_surrogate_id**(FK):foreign key to `DimPrizeRule`
- **tickets_bought**: number of tickets purchased in the transaction (up to 3)
- **sales**: total transaction amount
- **transaction_date**: actual date of transaction

Dimension Table: DimPrizeRule

- **prize_rule_surrogate_id**(PK): for the prize rule
- **winning_type_id** : Identifier for the winning condition
- **base_prize_fixed** : Base prize amount (unscaled)
- **base_prize_scaled** :Base prize amount after scaling (e.g., odds)
- **game_scratch_surrogate_id** (FK): foreign key to `DimScratchGame`
- **scratch_game_number**: identifier number of the scratch game
- **ticket_price**: standard ticket price for the game
- **average_odds**: average odds of winning in this game
- **average_odds_prob**: average probability associated with the odds
- **game_category**: category of the scratch game

- **game_type**: type or theme of the scratch game(scratch/non-scratch)

Dimension Table: DimScratchGame

- **game_scratch_surrogate_id** (PK): unique surrogate key for the scratch game
- **scratch_game_number**: identifier number of the scratch game
- **ticket_price**: standard ticket price for the game
- **average_odds**: average odds of winning in this game
- **average_odds_prob**: average probability associated with the odds
- **game_category**: category of the scratch game
- **game_type**: type or theme of the scratch game(scratch/non-scratch)

Dimension Table: DimDate

- **date_id_surrogate_id** (PK): unique surrogate key for the date
- **date_id**: unique date identifier (e.g., YYYYMMDD)
- **fiscal_year**: fiscal year number for the date
- **fiscal_month**: fiscal month number for the date
- **fiscal_month_name_and_number**: textual name and number of the fiscal month

Dimension Table: DimRetailer

- **retailer_license_number_surrogate_id** (PK): unique surrogate key for the retailer
- **retailer_license_number**: license number issued to the retailer
- **retailer_location_name**: name of the retailer location
- **retailer_number_and_location_name**: combined retailer number and location name for identification
- **retailer_location_zip_surrogate_id**(FK): foreign key to the zip address
- **retailer_location_zip_code**: ZIP code of the retailer location
- **retailer_location_city**: city of the retailer
- **retailer_location_state**: state of the retailer

Dimension Table: DimCustomer

- **customer_id_surrogate_id** (PK): surrogate key, unique identifier for the customer
- **customer_id**: original customer ID from source system

- **customer_name**: full name of the customer
- **customer_gender**: gender of the customer
- **customer_dob**: birth date of the customer
- **customer_email**: email address of the customer
- **customer_phone**: phone number of the customer
- **customer_registration_date**: date the customer registered
- **customer_location_zip_surrogate_id**(FK): surrogate key ZIP code
- **customer_zip**: ZIP code of residence
- **customer_city**: city of residence
- **customer_state**: state of residence

Dimension Table: DimEmployee

- **employee_id_surrogate_id** (PK): unique surrogate key for the employee
- **employee_id**: original employee ID
- **employee_name**: full name of the employee
- **employee_department**: department the employee belongs to
- **employee_email**: email address of the employee
- **employee_phone**: phone number of the employee
- **employee_hire_date**: hire date of the employee
- **employee_salary**: annual salary of the employee
- **employee_status**: current employment status (e.g., active, terminated)
- **retailer_license_number_surrogate_id** (FK): unique surrogate key for the retailer
- **retailer_license_number**: license number issued to the retailer
- **retailer_location_name**: name of the retailer location
- **retailer_number_and_location_name**: combined retailer number and location name for identification
- **retailer_location_zip_surrogate_id**(FK): primary street address of the retailer
- **retailer_location_zip_code**: ZIP code of the retailer location

- **retailer_location_city**: city of the retailer
- **retailer_location_state**: state of the retailer

1.4.2 Draw-Based Lottery

The dataset captures detailed information about draw-based lottery games. Unlike the previous scratch game, our assumption this time is that a player could win multiple times, as unlike the other one where his win is in the hands of the ticket seller, this time if he is sure of the right numbers (premonition), he could fill all three tickets and win big.

Fact Table: DrawTicketSalesFact

- **sale_id**(PK): unique identifier for the sales record
- **date_id_surrogate_id**(FK): foreign key to DimDate
- **game_id**(FK): foreign key to DimDrawGame
- **retailer_license_number_surrogate_id**(FK): foreign key to DimRetailer
- **customer_id_surrogate_id**(FK): foreign key to DimCustomer
- **employee_id_surrogate_id**(FK): foreign key to DimEmployee
- **chosen_numbers_combination_surrogate_id**(FK): foreign key to DimArrayCombinations
- **winning_numbers_combination_surrogate_id**(FK): foreign key to DimCombinations
- **ticket_price**: price of each ticket
- **sales**: total sale amount for the transaction
- **tickets_bought**: number of tickets purchased in transaction
- **transaction_date**: date and time when the transaction occurred
- **winning_ticket_count**: number of winning tickets in the transaction
- **is_winning_ticket**: boolean flag indicating whether the ticket is a winner
- **winning_prize**: prize tier/category won by the ticket
- **amount_won**: total monetary amount won in the transaction
- **game_draw_surrogate_id**(FK): surrogate key for game draw info

Dimension Table: DimDrawGame

- **game_id** (PK): original game identifier
- **game_category**: category or type of game (e.g., "million")
- **ticket_price**: default or standard price per ticket

Dimension Table: DimCombinations

- **winning_numbers_combination_surrogate_id** (PK): unique surrogate key for the combination
- **winning_combination_id**: original combination identifier
- **winning_numbers_combination**: the actual numbers combination
- **game_category**: category of the game associated with the combination

Dimension Table: DimArrayCombinations

- **chosen_numbers_combination_surrogate_id** (PK): unique surrogate key for the array of number combinations
- **game_category**
- **ticket_1_surrogate_id**(FK): foreign key to DimCombinations
- **ticket_1_combination_id**
- **ticket_1_combination_string**
- **ticket_2_surrogate_id**(FK): foreign key to DimCombinations
- **ticket_2_combination_id**
- **ticket_2_combination_string**
- **ticket_3_surrogate_id**(FK): foreign key to DimCombinations
- **ticket_3_combination_id**
- **ticket_3_combination_string**

Dimension Table: DimDate

- **date_id_surrogate_id** (PK): unique surrogate key for the date
- **date_id**: unique date identifier (e.g., YYYYMMDD)
- **fiscal_year**
- **fiscal_month**

- **fiscal_month_name_and_number**

Dimension Table: DimRetailer

- **retailer_license_number_surrogate_id** (PK): surrogate key for retailer license
- **retailer_license_number**
- **retailer_location_name**
- **retailer_number_and_location_name**
- **retailer_location_zip_code_surrogate_id**(FK)
- **retailer_location_zip_code**
- **retailer_location_city**
- **retailer_location_state**

Dimension Table: DimCustomer

- **customer_id_surrogate_id** (PK)
- **customer_id**
- **customer_name**
- **customer_gender**
- **customer_dob**
- **customer_email**
- **customer_phone**
- **customer_registration_date**
- **customer_zip_surrogate_id**(FK)
- **customer_zip**
- **customer_city**
- **customer_state**

Dimension Table: DimEmployee

- **employee_id_surrogate_id** (PK)
- **employee_id**
- **employee_name**

- **employee_department**
- **employee_email**
- **employee_phone**
- **employee_hire_date**
- **employee_salary**
- **employee_status**

1.4.3 Proposed Use Cases

- Detect top-performing game types and retailers across counties.
- Build dashboards to compare scratch vs. draw-based sales performance.
- Track monthly growth, returns, and adjustments to enhance planning.
- Identify underperforming games and regions needing marketing support.

Hierarchies

1.4.4 Dimension Hierarchies of scratch Based Game

Hierarchy: Date Dimension This hierarchy defines the fiscal time structure for temporal analysis and aggregation in the `DimDate` table.

- **Level 1: Fiscal Year** — `fiscal_year`
- **Level 2: Fiscal Month (Number)** — `fiscal_month`
- **Level 3: Fiscal Month (Name or Number)** — `fiscal_month_name_number`
- **Level 4: Date ID** — `date_id`

Hierarchy: Retailer Dimension This hierarchy organizes retailers geographically using attributes in the `DimRetailer` table.

- **Level 1: State** — `retailer_state`
- **Level 2: County** — `retailer_county`
- **Level 3: City** — `retailer_city`
- **Level 4: Zip Code** — `retailer_zip_code`
- **Level 5: Retailer Zip Code Surrogate ID** — `retailer_zip_code_surrogate_id`

Hierarchy: Customer Dimension This hierarchy structures customers demographically for behavioral analysis, using fields in the `DimCustomer` table.

Customer Dimension Hierarchy Levels

- **Level 1: State** — `customer_state`
- **Level 2: City** — `customer_city`
- **Level 3: ZIP Code** — `customer_zip`
- **Level 4: Customer Name** — `customer_name`
- **Level 5: Customer IDs**
  - `customer_surrogate_id` (PK)
  - `customer_id` (Source System ID)

1.4.5 Dimension Hierarchies for Draw-Based Lottery

Draw Game Dimension Hierarchy Levels

- **Level 1: Game Category** — `game_category` (defines the type of draw game)
- **Level 2: Game IDs**
  - `game_surrogate_id` (PK)
  - `game_id` (Original ID, defines the specific round)

Hierarchy of Chosen Number Combinations

- **Level 1: Tickets combination surrogate ids** —
  - `Ticket_1_surrogate_id`
  - `Ticket_2_surrogate_id`
  - `Ticket_3_surrogate_id`
- **Level 2: Individual surrogate id of the combinations of tickets**–`chosen_numbers_combination_id`

Hierarchy: Date Dimension This hierarchy defines the fiscal time structure for temporal analysis and aggregation in the `DimDate` table.

- **Level 1: Fiscal Year** — `fiscal_year`
- **Level 2: Fiscal Month (Number)** — `fiscal_month`
- **Level 3: Fiscal Month (Name or Number)** — `fiscal_month_name_number`

- **Level 4: Date ID** — `date_id`

Hierarchy: Retailer Dimension This hierarchy organizes retailers geographically using attributes in the `DimRetailer` table.

- **Level 1: State** — `retailer_state`
- **Level 2: County** — `retailer_county`
- **Level 3: City** — `retailer_city`
- **Level 4: Zip Code** — `retailer_zip_code`
- **Level 5: Retailer Zip Code Surrogate ID** — `retailer_zip_code`$_s$*urrogate_id*

Hierarchy: Customer Dimension This hierarchy structures customers demographically for behavioral analysis, using fields in the `DimCustomer` table.

Customer Dimension Hierarchy Levels

- **Level 1: State** — `customer_state`
- **Level 2: City** — `customer_city`
- **Level 3: ZIP Code** — `customer_zip`
- **Level 4: Customer Name** — `customer_name`
- **Level 5: Customer IDs**
  - `customer_surrogate_id` (PK)
  - `customer_id` (Source System ID)

1.5 Grain. DIM. FACT

1.5.1 Scratch Based Game

Column Name	PostgreSQL Type	Example	Description
sale_id	BIGINT	1	Unique identifier for each sale record
date_id_surrogate_id	BIGINT	240	Surrogate key for date
game_scratch_surrogate_id	BIGINT	3	Surrogate key for scratch game
retailer_license_number_surrogate_id	BIGINT	28	Surrogate key for retailer license number
customer_id_surrogate_id	BIGINT	12463	Surrogate key for customer
employee_id_surrogate_id	BIGINT	4580	Surrogate key for employee

prize_rule_surrogate_id	BIGINT	4	Surrogate key for prize rule
tickets_bought	BIGINT	1	Number of tickets bought
sales	BIGINT	30	Total sales amount
transaction_date	DATE	2021-03-25	Date of transaction
winning_type_id	VARCHAR(50)	none	Identifier for winning type
base_prize_fixed	BIGINT	0	Fixed base prize amount
base_prize_scaled	BIGINT	0	Scaled base prize amount
scratch_game_number	INTEGER	2053	Identifier for the scratch game
ticket_price	INTEGER	30	Price per ticket
average_odds	VARCHAR(20)	1:3.44	Average odds as ratio
average_odds_prob	NUMERIC(6,4)	0.2907	Probability of average odds
game_category	VARCHAR(100)	Scratch Tickets	Category of the game
game_type	VARCHAR(50)	Scratch	Type of the game
date_id	BIGINT	20210325	Numeric date identifier (YYYYMMDD)
fiscal_year	SMALLINT	2021	Fiscal year of the transaction
fiscal_month	SMALLINT	7	Fiscal month number
fiscal_month_name_and_number	VARCHAR(20)	07-March	Fiscal month as number and name
retailer_license_number	BIGINT	100663	Retailer license number
retailer_location_name	VARCHAR(100)	ISI-KAT	Name of retailer location
retailer_number_and_location_name	VARCHAR(150)	100663 - ISI-KAT	Retailer number and location name
retailer_location_zip_code_surrogate_id	BIGINT	9989	Surrogate ID for retailer zip code
retailer_location_zip_code	BIGINT	78212	Retailer zip code
retailer_location_city	VARCHAR(100)	San Antonio	Retailer city
retailer_location_state	VARCHAR(5)	TX	Retailer state code
customer_id	VARCHAR(50)	SAN ANTONIO_137	Unique customer identifier
customer_name	VARCHAR(100)	Matthew Harris	Customer full name
customer_gender	VARCHAR(10)	M	Customer gender
customer_dob	DATE	2021-03-28	Customer date of birth
customer_email	VARCHAR(150)	heatherlam@hotmail.com	Customer email address
customer_phone	VARCHAR(30)	+1-819-675-8143x61145	Customer phone number
customer_registration_date	DATE	2015-07-04	Customer registration date
customer_zip	BIGINT	60432	Customer postal code
customer_city	VARCHAR(100)	San Antonio	Customer city
customer_state	VARCHAR(5)	TX	Customer state code
employee_id	VARCHAR(50)	EMP_100663_3	Employee unique identifier
employee_name	VARCHAR(100)	John Waters	Employee full name
employee_department	VARCHAR(100)	Sales Associate	Employee department
employee_email	VARCHAR(150)	anthony89@example.org	Employee email address

employee_phone	VARCHAR(30)	448-544-2941	Employee phone number
employee_hire_date	DATE	2016-02-09	Employee hire date
employee_salary	BIGINT	34535	Employee salary
employee_status	VARCHAR(20)	Inactive	Employee status
customer_zip_surrogate_id	BIGINT	7330	Surrogate ID for customer zip

1.5.2 Draw Based Game

Column Name	PostgreSQL Type	Example	Description
sale_id	BIGINT	6	Unique identifier for each sale record
date_id_surrogate_id	BIGINT	179	Surrogate key for date
game_id	VARCHAR(100)	all_or_nothing_05_january_100663	Unique game identifier
retailer_license_number_surrogate_id	BIGINT	28	Surrogate key for retailer license
customer_id_surrogate_id	BIGINT	12327	Surrogate key for customer
employee_id_surrogate_id	BIGINT	4578	Surrogate key for employee
chosen_numbers_combination_surrogate_id	BIGINT	6	Surrogate key for chosen numbers combination
winning_numbers_combination_surrogate_id	BIGINT	4	Surrogate key for winning numbers combination
ticket_price	INTEGER	1	Price per ticket
sales	BIGINT	3	Total sales amount
tickets_bought	BIGINT	3	Number of tickets bought
transaction_date	DATE	2021-01-12	Date of transaction
winning_ticket_count	BIGINT	0	Number of winning tickets
is_winning_ticket	BOOLEAN	False	Whether the ticket is winning
winning_prize	BIGINT	0	Winning prize amount
amount_won	BIGINT	0	Amount won
game_draw_surrogate_id	BIGINT	50	Surrogate key for game draw
game_category	VARCHAR(100)	All or Nothing	Category of game
winning_combination_id	VARCHAR(50)	ALLORNOTHING_000004	Identifier for winning combination
winning_numbers_combination	VARCHAR(150)	2-4-5-9-12-13-15-19-20-21-23-24	Winning numbers combination string
chosen_ids	JSONB	['ALLORNOTHING_000011', 'ALLORNOTHING_000012', 'ALLORNOTHING_000013']	List of chosen IDs

chosen_numbers_combination	JSONB	['2-3-4-5-6-8-9-10-13-14-17-24', '1-3-4-7-8-9-10-11-18-21-23-24', '1-3-4-6-7-9-10-13-17-20-21-22']	List of chosen numbers combinations
chosen_numbers_combination_surrogate_ids	JSONB	[11, 12, 13]	Surrogate IDs for chosen numbers combinations
ticket_1_surrogate_id	BIGINT	11	Surrogate key for ticket 1
ticket_1_combination_id	VARCHAR(50)	ALLORNOTHING_-000011	Combination ID for ticket 1
ticket_1_combination_string	VARCHAR(150)	2-3-4-5-6-8-9-10-13-14-17-24	Combination string for ticket 1
ticket_2_surrogate_id	BIGINT	20	Surrogate key for ticket 2
ticket_2_combination_id	VARCHAR(50)	ALLORNOTHING_-000020	Combination ID for ticket 2
ticket_2_combination_string	VARCHAR(150)	2-4-7-8-11-14-16-17-19-21-23-24	Combination string for ticket 2
ticket_3_surrogate_id	BIGINT	35	Surrogate key for ticket 3
ticket_3_combination_id	VARCHAR(50)	ALLORNOTHING_-000035	Combination ID for ticket 3
ticket_3_combination_string	VARCHAR(150)	2-5-6-12-13-14-17-18-19-21-23-24	Combination string for ticket 3
date_id	BIGINT	20210112	Numeric date identifier (YYYYMMDD)
fiscal_year	SMALLINT	2021	Fiscal year of transaction
fiscal_month	SMALLINT	5	Fiscal month number
fiscal_month_name_and_-number	VARCHAR(20)	05-January	Fiscal month as number and name
retailer_license_number	BIGINT	100663	Retailer license number
retailer_location_name	VARCHAR(100)	ISI-KAT	Retailer location name
retailer_number_and_location_name	VARCHAR(150)	100663 - ISI-KAT	Retailer number and location name
retailer_location_zip_code_-surrogate_id	BIGINT	8951	Surrogate key for zip code
retailer_location_zip_code	BIGINT	78212	Zip code
retailer_location_city	VARCHAR(100)	San Antonio	City
retailer_location_state	VARCHAR(5)	TX	State code
customer_id	VARCHAR(50)	SAN ANTONIO_137	Customer ID
customer_name	VARCHAR(100)	Matthew Harris	Customer name
customer_gender	VARCHAR(10)	M	Gender
customer_dob	DATE	2021-01-25	Date of birth
customer_email	VARCHAR(150)	matthew.harris@example.com	Email
customer_phone	VARCHAR(30)	+1-111-111-1111	Phone
customer_registration_date	DATE	2015-07-04	Registration date
customer_zip	BIGINT	60432	Postal code
customer_city	VARCHAR(100)	San Antonio	City

customer_state	VARCHAR(5)	TX	State
employee_id	VARCHAR(50)	EMP_100663_3	Employee ID
employee_name	VARCHAR(100)	John Waters	Employee name
employee_department	VARCHAR(100)	Sales Associate	Department
employee_email	VARCHAR(150)	anthony89@example.org	Email
employee_phone	VARCHAR(30)	448-544-2941	Phone
employee_hire_date	DATE	2016-02-09	Hire date
employee_salary	BIGINT	34535	Salary
employee_status	VARCHAR(20)	Inactive	Status
customer_zip_surrogate_id	BIGINT	7330	Customer zip surrogate

1.5.3 Grain Selection Process

To ensure our fact tables accurately reflect the business processes of the Texas Lottery system, we have followed the classic four-step method for defining the grain of a fact table, as outlined by Ralph Kimball. The essence of this approach lies in performing the hard work upfront—carefully identifying the highest-level variables to include in each fact table—so that querying becomes significantly easier later. By distilling the fact tables to their most granular, meaningful level and linking them to relevant dimension tables, we enable streamlined and intuitive analytics.

1. Choose the Business Process

We began by identifying the key business processes we aim to model:

- **Scratch-Based Lottery Sales:** Instant-win ticket purchases where outcomes are known at the time of purchase.
- **Draw-Based Lottery Sales:** Number-selection games where outcomes are determined in a later draw event.

These two distinct processes led us to create two separate fact tables: `ScratchTicketSalesFact` and `DrawTicketSalesFact`.

2. Declare the Grain

The grain defines the level of detail represented by each row in the fact table:

- For `ScratchTicketSalesFact`, the grain is **one scratch ticket sales transaction**—e.g., a customer purchasing 3 tickets for **Game #2085: $500 Frenzy** in a single transaction.
- For `DrawTicketSalesFact`, the grain is **one draw-based ticket transaction**, capturing selected numbers and the corresponding winning combinations.

Clearly declaring the grain has guided our decisions about which dimensions and facts are appropriate for each table.

3. Identify the Dimensions

Next, we determined the necessary dimension tables to describe the context surrounding each fact:

- `ScratchTicketSalesFact` is linked to: `DimScratchGame`, `DimRetailer`, `DimCustomer`, `DimEmployee`, and `DimDate`.
- `DrawTicketSalesFact` connects to: `DimDrawGame`, `DimCombinations`, `DimArrayCombinations`, `DimRetailer`, `DimCustomer`, `DimEmployee`, and `DimDate`.

These dimensions enable rich analysis by game type, customer demographics, retailer performance, employee involvement, and time. For draw-based games, additional dimensions like `DimCombinations` and `DimArrayCombinations` support detailed tracking of selected and winning number combinations.

4. Determine the Facts

Finally, we identified the quantitative measures that are meaningful at each grain level:

- `ScratchTicketSalesFact`: `ticket_price`, `tickets_bought`, and `sale` (e.g., 3 tickets at $5 = $15).
- `DrawTicketSalesFact`: `ticket_price`, `tickets_bought`, `ticket_sales`, `winning_ticket_count`, `is_winning_ticket`, `winning_prize`, and `amount_won`.

These facts enable comprehensive insights into customer purchasing behavior, game profitability, and overall revenue trends.

2 Business Layer 3NF

The design process has started with a thorough analysis of the initial star schemas for both the Scratch-Based and Draw-Based lottery systems. The primary objective has been to transform these into a unite, normalized schema in 3rd Normal Form (3NF), ensuring minimal redundancy, improved data integrity, and clear separation of concerns between facts and dimensions.

1. Identifying Common and Specific Dimensions

First, I have spotted the entities that were common across both lottery types, as well as those specific to each. This helped in determining which tables could be shared as global dimensions and which needed to remain source-specific.

- **Shared Dimensions:** `DimCustomer`, `DimDate`, `DimRetailer`, and `DimEmployee` are present in both systems and thus are designed as central dimension tables.
- **Scratch-Specific:** Entities such as `DimPrizeRule` and `DimScratchGame` are specific to the scratch-based system only.
- **Draw-Specific:** Entities like `DimDrawGame`, `DimArrayCombinations`, and `DimCombinations` are unique to the draw-based system.

2. Designing Dimension Tables in 3NF

Once the relevant dimensions are identified, I have normalized each into 3NF. This involved eliminating transitive dependencies, ensuring atomicity, and structuring each dimension table with a surrogate primary key for efficient joins.

- `DimCustomer` has been normalized to include only customer-specific attributes (e.g., name, address, contact information).
- `DimRetailer` hes been separated from employee details, with `DimEmployee` created to hold employee-specific data, linked via a foreign key to the retailer.
- `DimDate` has been shared across all fact tables for consistency in date-related analysis.

3. Fact Tables

Separate fact tables have been maintained for each lottery system, with each table referencing the relevant dimension tables through foreign keys.

- `FactScratchSales` captures transactions related to scratch ticket purchases, linking to dimensions such as customer, date, retailer, employee, prize rule, and scratch game.
- `FactDrawSales` logs draw-based ticket sales, referencing dimensions including customer, date, retailer, draw game, array combinations, and combinations.

Key measures—such as `amount_spent`, `ticket_count`, and `winning_amount`—are included as quantitative metrics to represent sales data accurately. In addition, attributes like `is_winning`, `combination_winning`, and `base_prize_scaled` are used to characterize winning outcomes, which will later enable the calculation of net sales per customer. At this stage, only gross sales are available, but the structure anticipates real-world scenarios in which retailers must account for prize payouts.

4. Ensuring Referential Integrity and Flexibility

All dimension tables have been assigned surrogate keys (e.g., `customer_id_surrogate_id`, `retailer_id_surrogate_id`) to standardize relationships and improve query performance. Foreign key constraints have been established to ensure referential integrity across the schema.

5. Final Integration

In the end, the full schema has been evaluated to ensure consistency, normalization, and scalability. By maintaining a clear separation between dimensions and facts and aligning both lottery systems under shared structures, the resulting 3NF schema supports efficient querying and future extensibility.

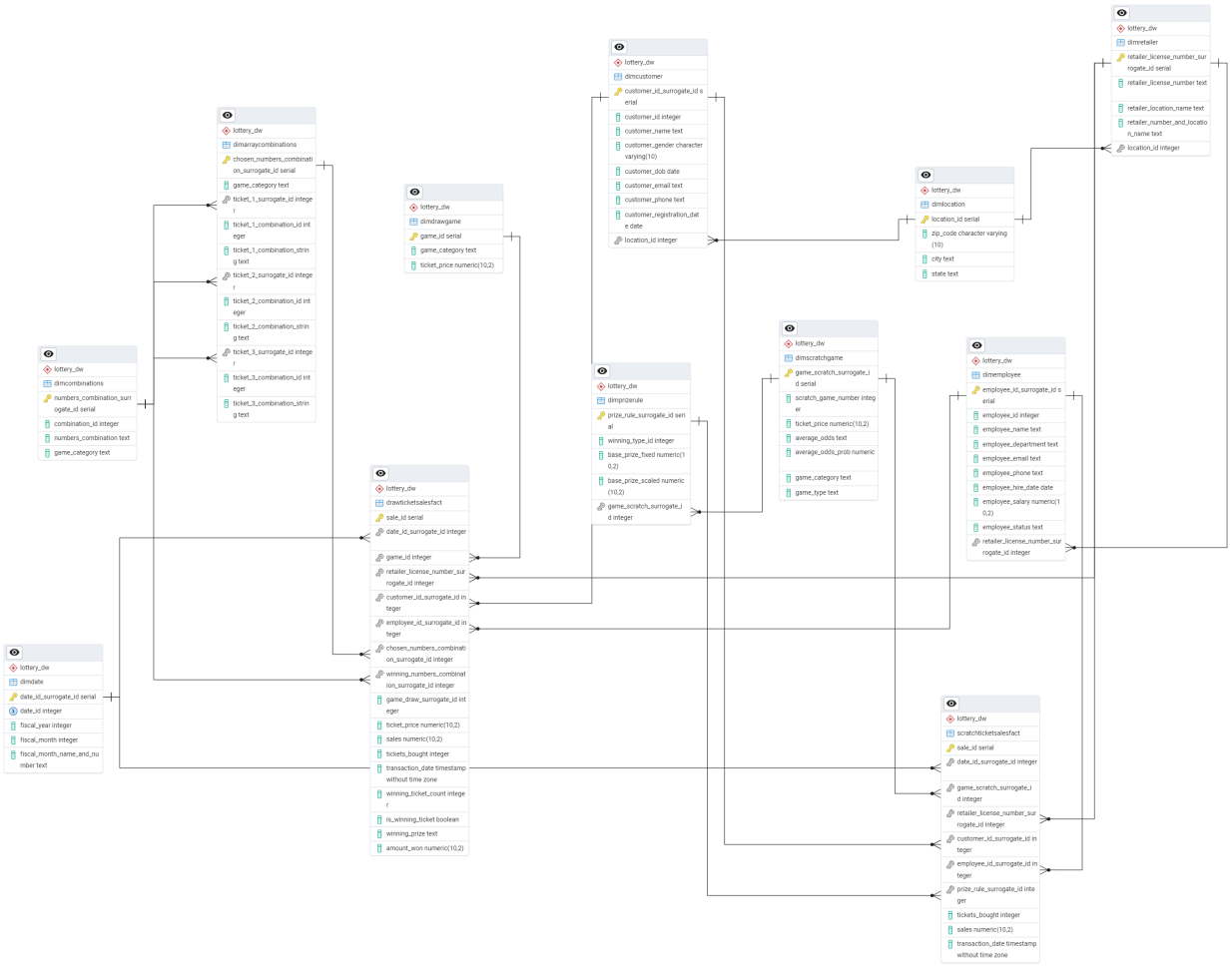

Figure 1: 3NF Modelling Of The Lottery Database