Assignment 1 - Part 1: Normalization Analysis

Student: Payal Sharma
Dataset: techcrunch.csv

Question 1: Primary Key Selection (10 pts)

Good Choice for Primary Key: `fund_id`

Reasoning:

- The `fund_id` column contains unique values (1-523) with no duplicates
- Each fund_id uniquely identifies a single funding event
- It's a simple, single-column key that is easy to reference
- It satisfies both properties of a primary key:
  - **Uniqueness**: No two rows have the same fund_id
  - **Non-null**: Every row has a fund_id value

Invalid Primary Key Examples:

Example: Composite Key of (`company`, `fundedDate`) (Invalid)

- **Why it fails:** A company can receive multiple funding rounds on the same date
- **Evidence:** Grouply received two angel rounds on 1/1/2008 (fund_id 143, 144)
- **Violates:** Uniqueness requirement even with two columns combined

---

Question 2: First Normal Form (1NF) (5 pts)

Answer: YES, the table satisfies 1NF

Criteria for 1NF:

1. All columns contain atomic (indivisible) values
2. Each column contains values of a single data type
3. Each column has a unique name
4. The order of rows doesn't matter

Analysis:

i) **Atomic Values:**

- Every cell contains a single, indivisible value
- No repeating groups or arrays within cells
- Examples:
  - `company`: "Facebook" (single text value)
  - `numEmps`: 450 (single numeric value)
  - `raisedAmt`: 500000 (single numeric value)

ii) **Single Data Type per Column:**

iii) **Unique Column Names:**

- All 10 columns have distinct names with no duplicates

iv) **Row Order Independence:**

- Data is meaningful regardless of row ordering
- Each row is identified by fund_id

Conclusion: The techcrunch.csv dataset is in 1NF because all values are atomic, columns have consistent data types, and there are no repeating groups.

Question 3: Second Normal Form (2NF) (5 pts)

Answer: NO, the table does NOT satisfy 2NF

Criteria for 2NF:

1. Must be in 1NF (satisfied, as shown above)
2. All non-key attributes must be fully functionally dependent on the entire primary key (no partial dependencies)

Analysis:

Since our primary key is `fund_id` (a single column), there cannot be partial dependencies in the traditional sense. However, examining the data reveals **transitive dependencies** and **redundancy** that violate the spirit of normalization:

Evidence of 2NF Violations:

i) The same company information is repeated across multiple rows:
   - **Facebook** (fund_id 8-14): `numEmps=450`, `category=web`, `city=Palo Alto`, `state=CA` repeated 7 times

ii) **Functional Dependencies:** The following attributes are functionally dependent on `company`, NOT on `fund_id`:

- `company` → `numEmps` (number of employees is a property of the company)
- `company` → `category` (business category is a property of the company)
- `company` → `city` (headquarters location is a property of the company)

iii) **Update Anomalies:**

- **Insertion Anomaly:** Cannot add a new company without a funding event
- **Update Anomaly:** If Facebook moves offices or changes employee count, we must update 7 rows
- **Deletion Anomaly:** Deleting all funding events for a company also deletes the company information

Conclusion: The table is NOT in 2NF because company attributes (numEmps, category, city, state) are repeated for each funding event and should be separated into a distinct Company table.

Question 4: Third Normal Form (3NF) (5 pts)

Answer: NO, the table does NOT satisfy 3NF

Criteria for 3NF:

1. Must be in 2NF (NOT satisfied, as shown above)
2. No transitive dependencies (all non-key attributes must depend directly on the primary key, not on other non-key attributes)

Analysis:

Since the table fails 2NF, it automatically fails 3NF.

Conclusion: The table is NOT in 3NF due to:

1. Failure to meet 2NF (prerequisite for 3NF)
2. Transitive dependencies where company attributes depend on company name, not directly on fund_id
3. Significant data redundancy causing update, insertion, and deletion anomalies

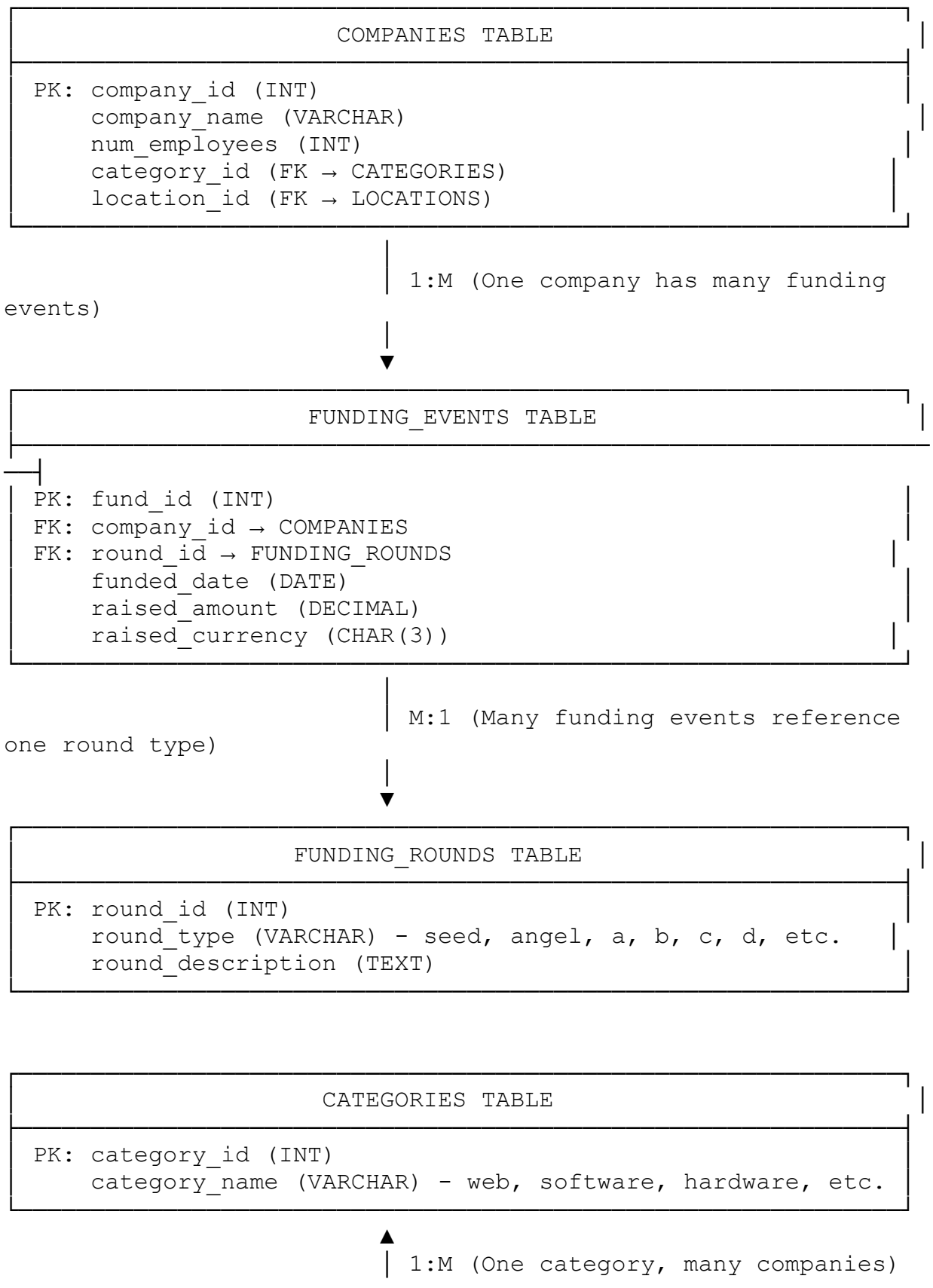
---

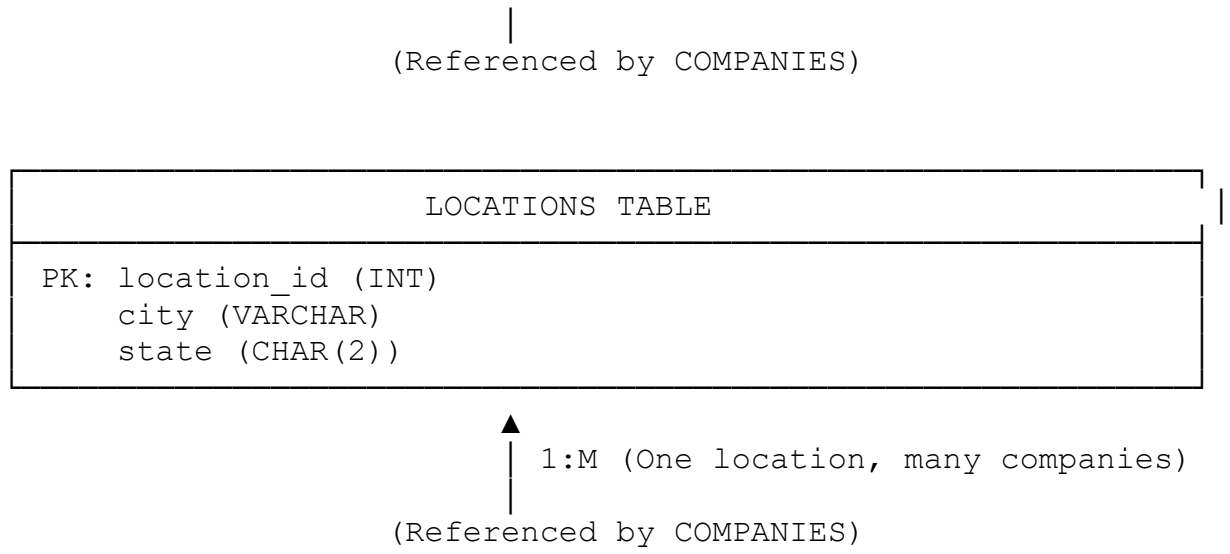
Question 5: Entity-Relationship Diagram for 3NF (25 pts)

Proposed Normalized Database Design

To bring the techcrunch.csv dataset into 3NF, we need to decompose it into multiple tables to eliminate redundancy and transitive dependencies.

Normalized Table Structure:


COMPANIES TABLE
PK: company_id (INT)
company_name (VARCHAR)
num_employees (INT)
category_id (FK → CATEGORIES)
location_id (FK → LOCATIONS)
1:M (One company has many funding events)
FUNDING_EVENTS TABLE
PK: fund_id (INT)
FK: company_id → COMPANIES
FK: round_id → FUNDING_ROUNDS
funded_date (DATE)
raised_amount (DECIMAL)
raised_currency (CHAR(3))
M:1 (Many funding events reference one round type)
FUNDING_ROUNDS TABLE
PK: round_id (INT)
round_type (VARCHAR) - seed, angel, a, b, c, d, etc.
round_description (TEXT)
CATEGORIES TABLE
PK: category_id (INT)
category_name (VARCHAR) - web, software, hardware, etc.
1:M (One category, many companies)



Relationship Details:

1. COMPANIES ↔ FUNDING_EVENTS

- **Type:** One-to-Many (1:M)
- **Description:** One company can have multiple funding events over time
- **Example:** Facebook (company_id=1) has 7 funding events (fund_id 8-14)

2. CATEGORIES ↔ COMPANIES

- **Type:** One-to-Many (1:M)
- **Description:** One category contains many companies, but each company belongs to one category
- **Example:** "web" category includes Facebook, Twitter, Digg, etc.

3. LOCATIONS ↔ COMPANIES

- **Type:** One-to-Many (1:M)
- **Description:** One location (city-state pair) can have multiple companies, but each company has one headquarters location
- **Example:** "San Francisco, CA" is home to Twitter, Scribd, Mogulus, and many others

4. FUNDING_ROUNDS ↔ FUNDING_EVENTS

- **Type:** One-to-Many (1:M)
- **Description:** One round type is used by many funding events
- **Example:** "Series A" round type is used for hundreds of different funding events across companies

---

Why This Design Achieves 3NF:

i) **Eliminates Redundancy:**
ii) **Satisfies 1NF:**
iii) **Satisfies 2NF:**
iv) **Satisfies 3NF:**
v) **Prevents Anomalies:**