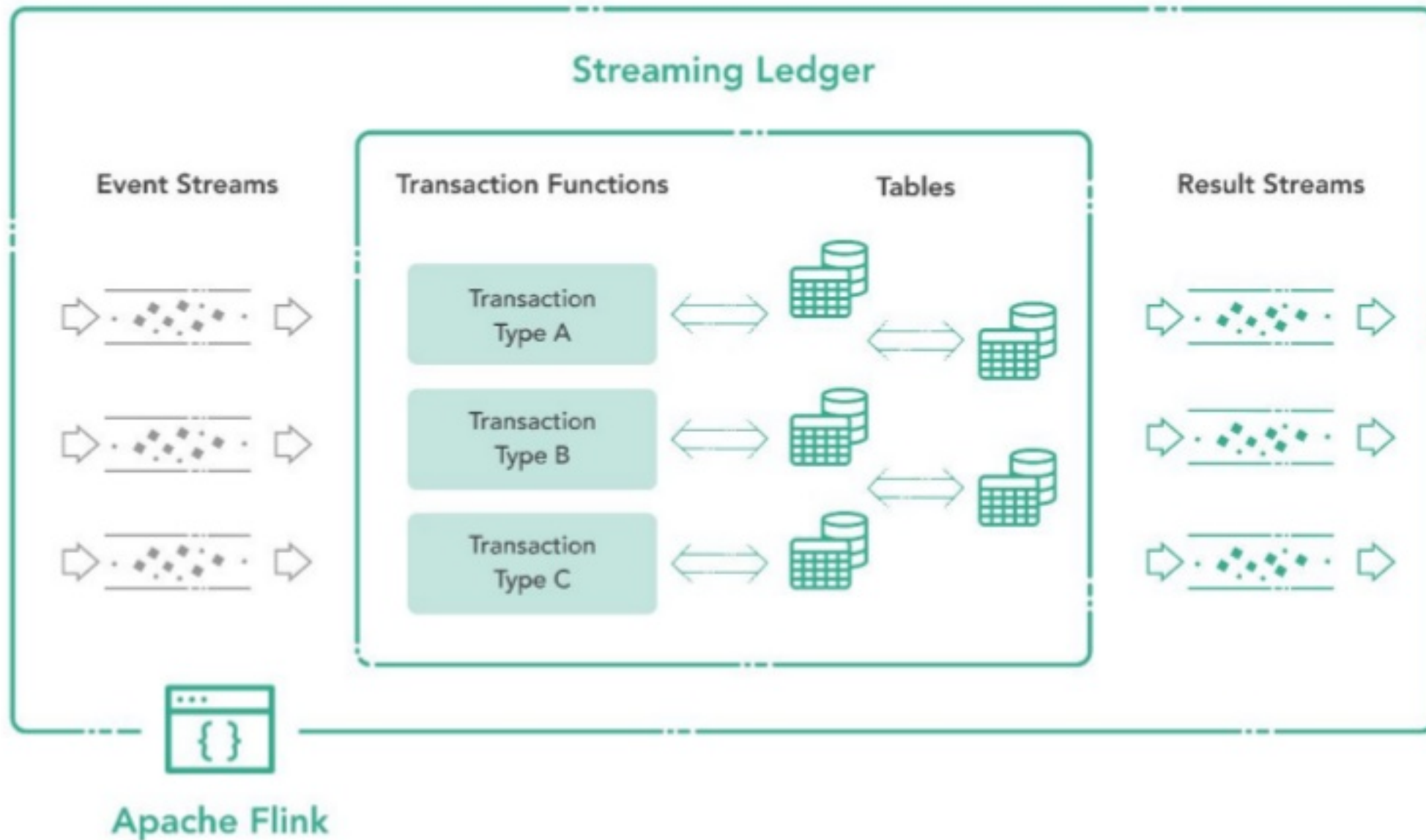# data Artisans Streaming Ledger

Serializable ACID transactions on streaming data

data Artisans Streaming Ledger

MOTIVATING EXAMPLE

MOTIVATING EXAMPLE

A financial service

- Accounts identified by an id, with a balance.

MOTIVATING EXAMPLE

A financial service

- Accounts identified by an id with a balance.

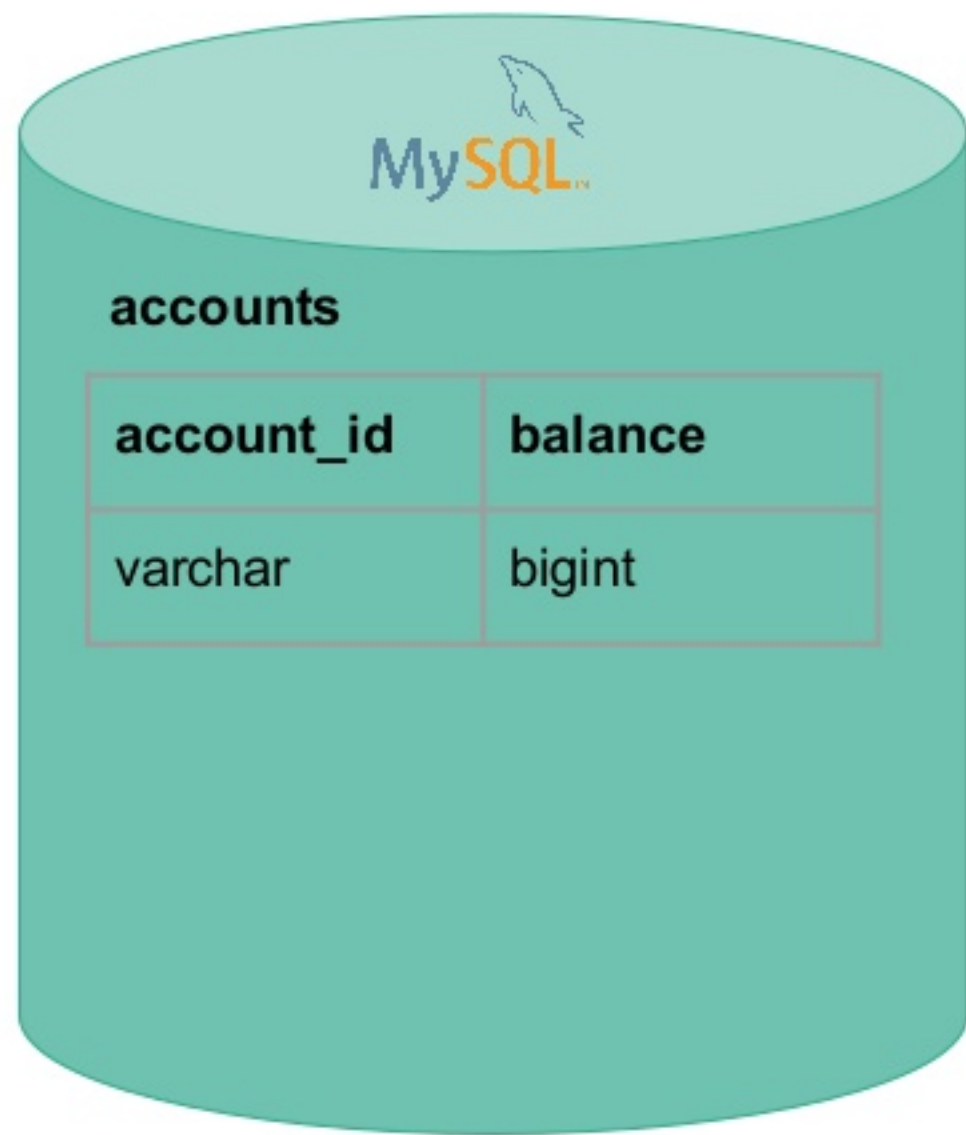- Deposit 💰 to an account

MOTIVATING EXAMPLE

A financial service

- Accounts identified by an id with a balance.

- Deposit 💰 to an account

- Find out the current balance of an account
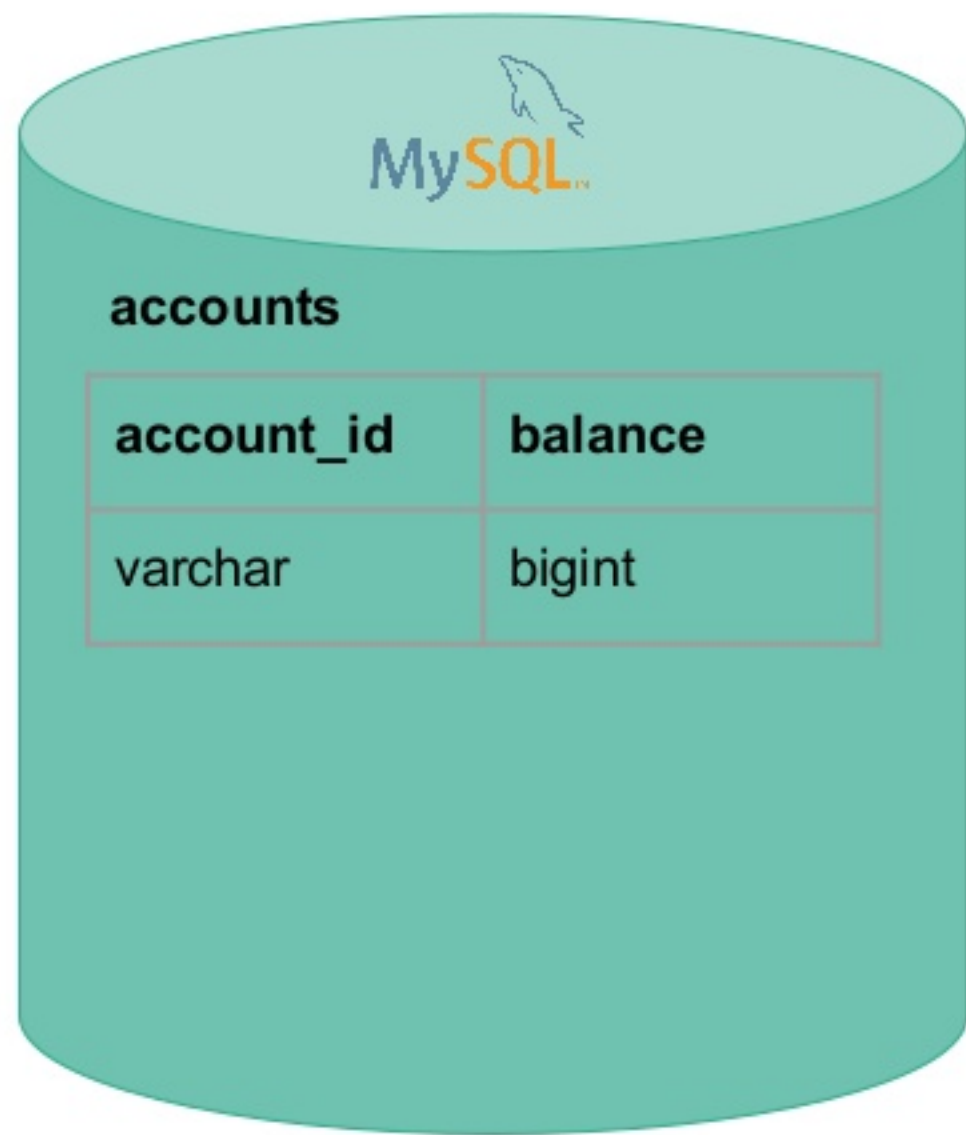
MOTIVATING EXAMPLE

A financial service

- Accounts identified by an id with a balance.

- Deposit 💰 to an account

- Find out the current balance of an account
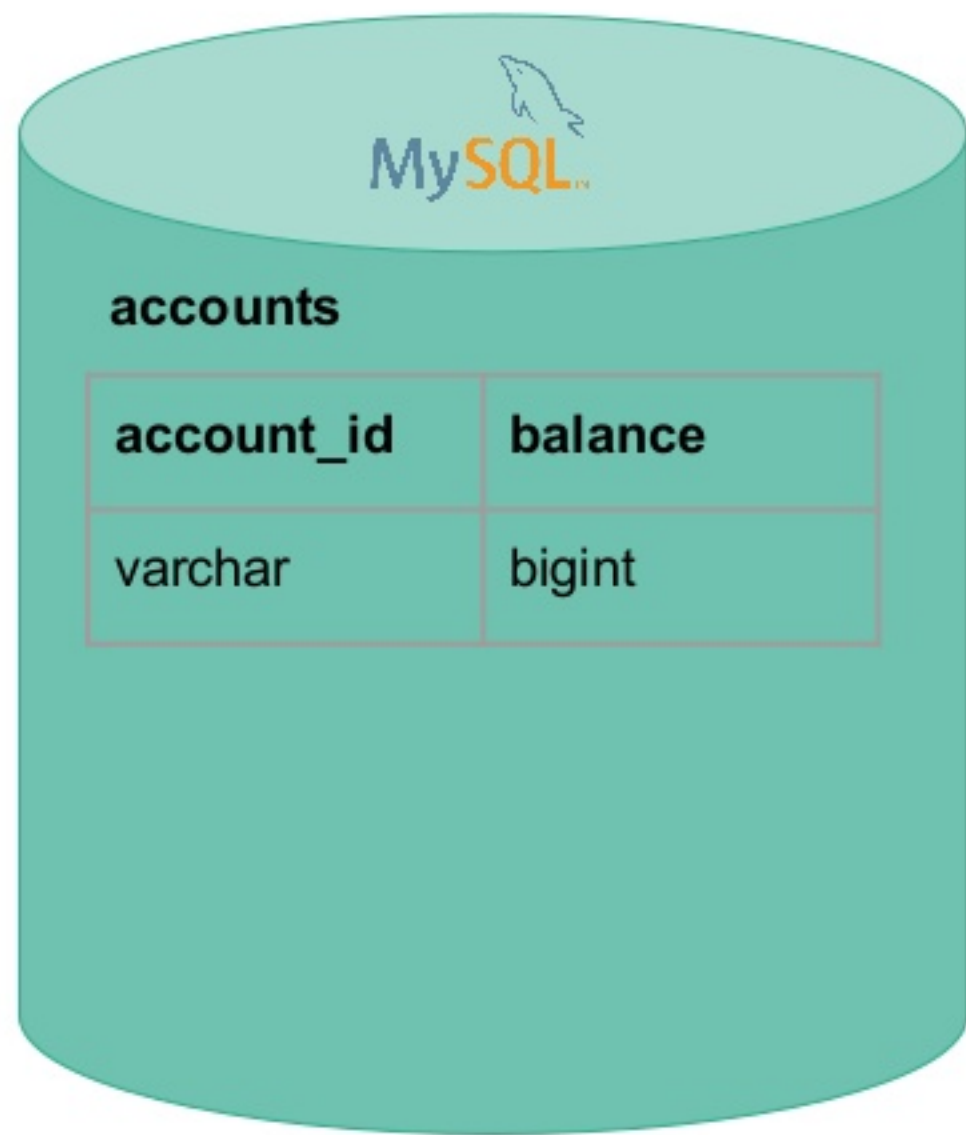
- Transfer 💰 between accounts

MOTIVATING EXAMPLE

accounts

account_id	balance
varchar	bigint

MOTIVATING EXAMPLE

accounts

account_id	balance
varchar	bigint

MOTIVATING EXAMPLE


```
INSERT INTO accounts (account_id, balance)
VALUES (?, ?);
```
Deposit

```
SELECT balance FROM accounts

  WHERE account_id = ?
```
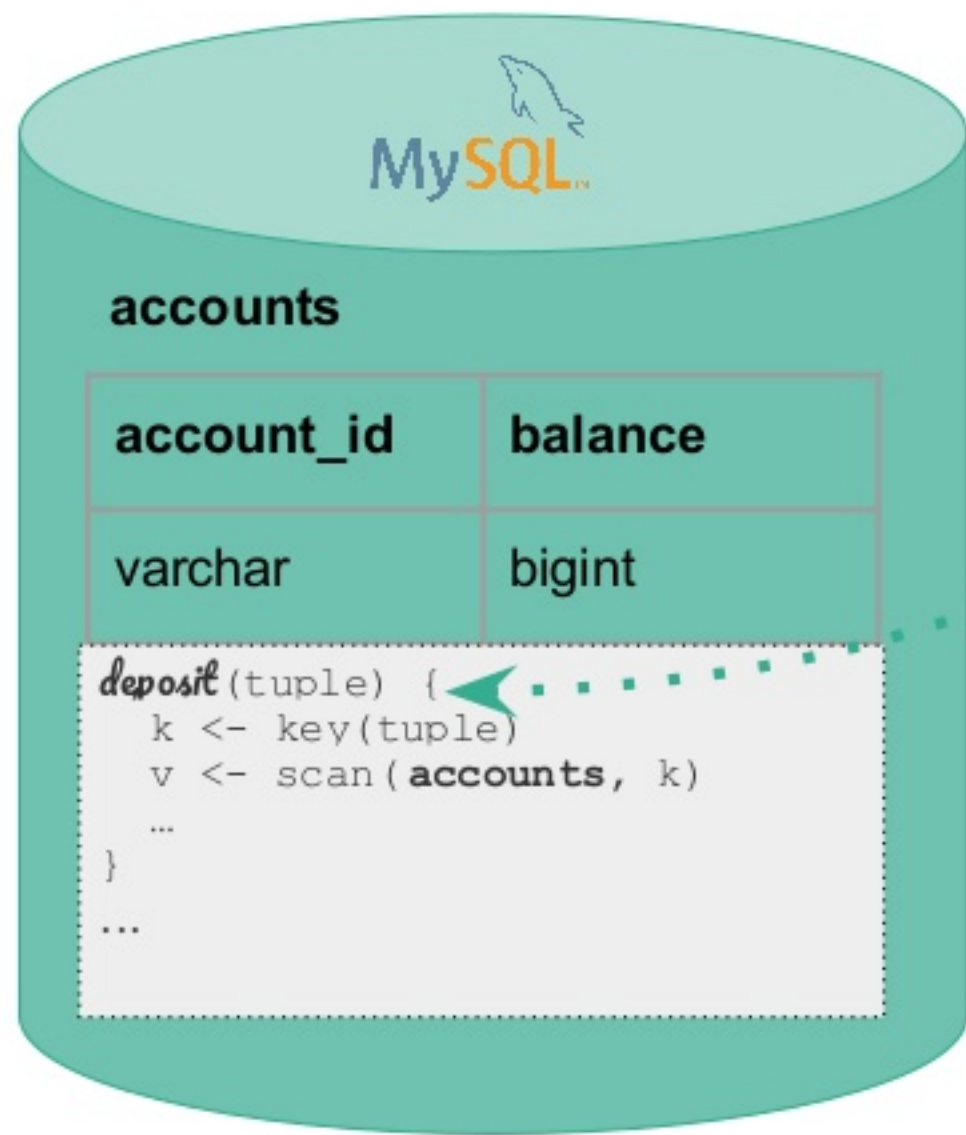*Balance inquiry*

```
UPDATE accounts
SET balance = balance - ?
WHERE account_id = ? ;

UPDATE accounts
SET balance = balance + ?
WHERE account_id = ?
```
Transfer

MOTIVATING EXAMPLE

accounts

account_id	balance
varchar	bigint

```
deposit(tuple) {
  k <- key(tuple)
  v <- scan(accounts, k)
  …
}
…
```

- ```sql
  INSERT INTO accounts (account_id, balance)
  VALUES (?, ?);
  ```

  Deposit
  `(varchar, bigint)`

- ```sql
  SELECT balance FROM accounts
   WHERE account_id = ?
  ```

  Balance inquiry
  `(varchar)`

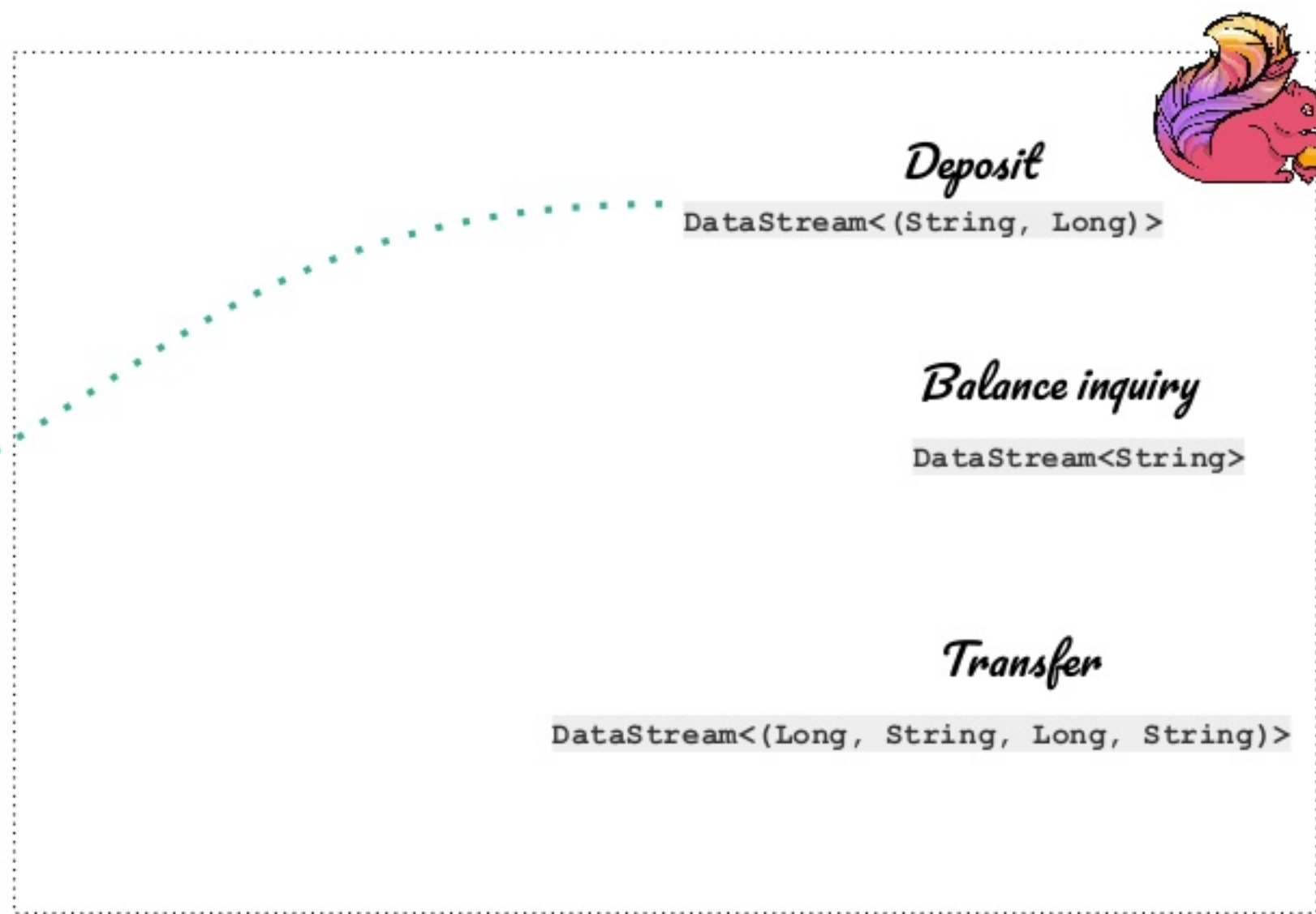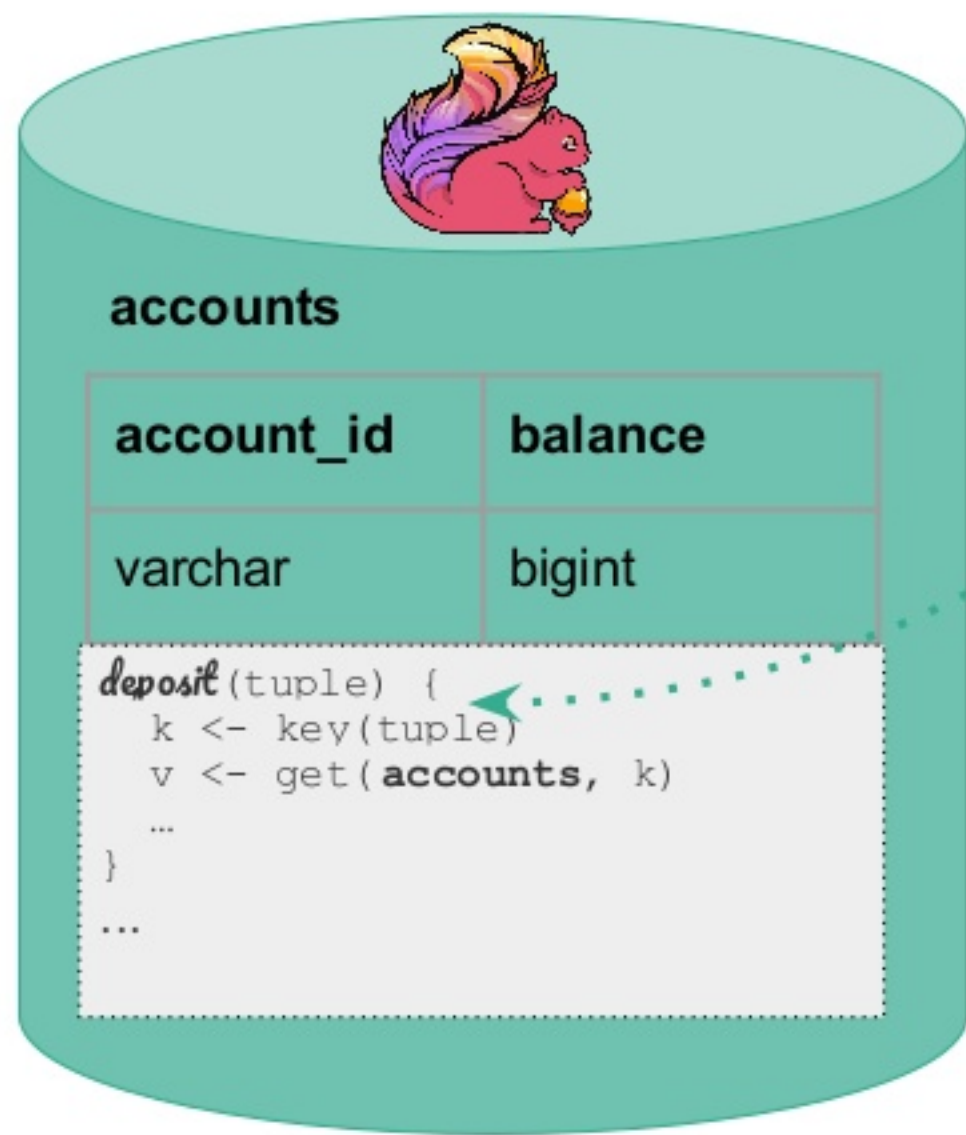- ```sql
  UPDATE accounts
  SET balance = balance - ?
  WHERE account_id = ? ;

  UPDATE accounts
  SET balance = balance + ?
  WHERE account_id = ?
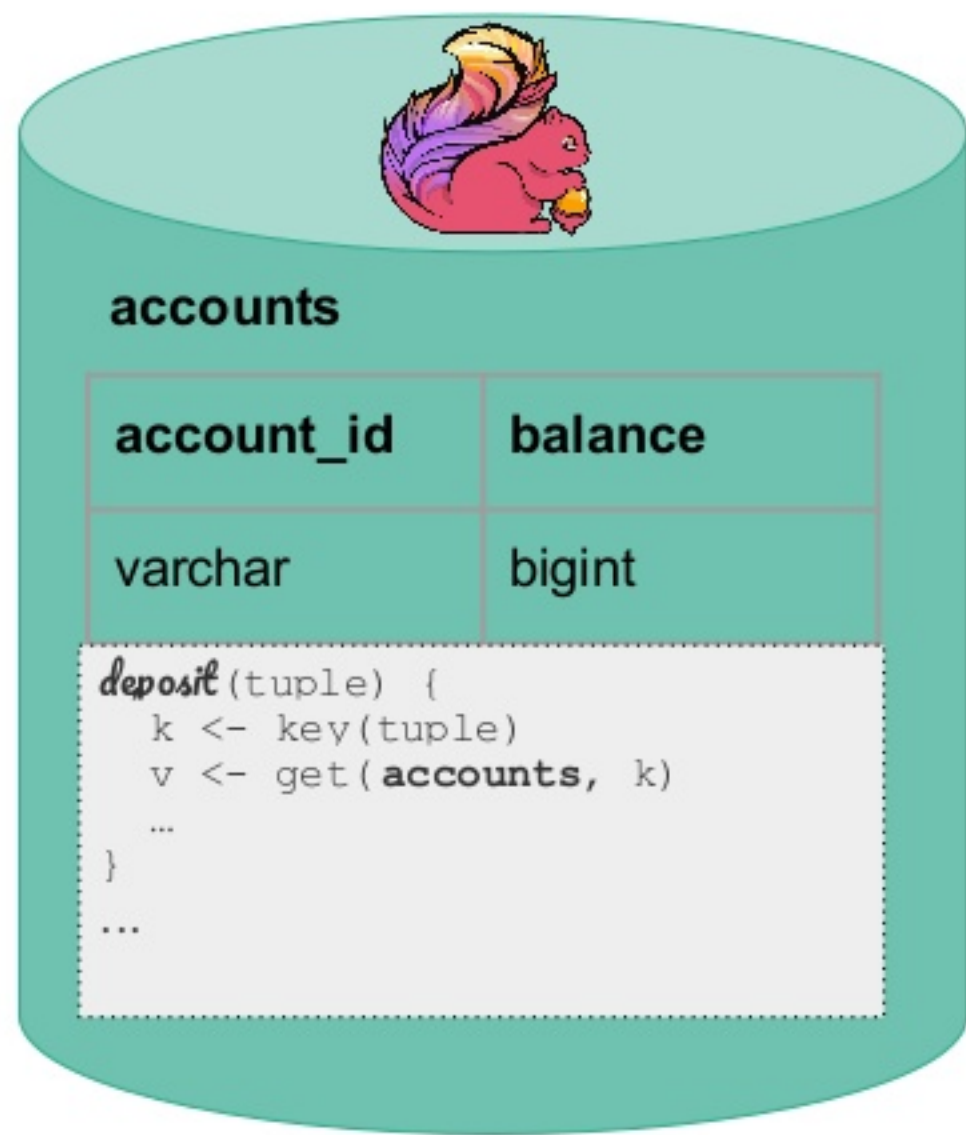 ```

  Transfer
  `(bigint, varchar, bigint, varchar)`

MOTIVATING EXAMPLE

accounts

account_id	balance
varchar	bigint

```
deposit(tuple) {
  k <- key(tuple)
  v <- get(accounts, k)
  ...
}
...
```

Deposit

`DataStream<(String, Long)>`

Balance inquiry

`DataStream<String>`

Transfer

`DataStream<(Long, String, Long, String)>`

MOTIVATING EXAMPLE

accounts

account_id	balance
varchar	bigint

```
deposit(tuple) {
  k <- key(tuple)
  v <- get(accounts, k)
  …
}
…
```

- No Intra-Key state access

- No transactions across multiple keys.

data Artisans

STREAMING LEDGER

A LIBRARY ON TOP OF APACHE FLINK

- No additional dependencies needed

- Seamlessly integrates and composes with DataStream API and SQL

- Read from- and write to all Flink connectors

- Supports savepoints for upgrades

data Artisans

STREAMING LEDGER

- Walkthrough

GETTING STARTED

https://github.com/dataArtisans/da-streamingledger

dA

PLATFORM

https://data-artisans.com/platform-overview

Part of data Artisans Platform River Edition

Streaming Ledger will be released as part of data Artisans Platform

- **Stream edition**: Flink, Application Manager

- **River edition**: Flink, Application Manager, Streaming Ledger
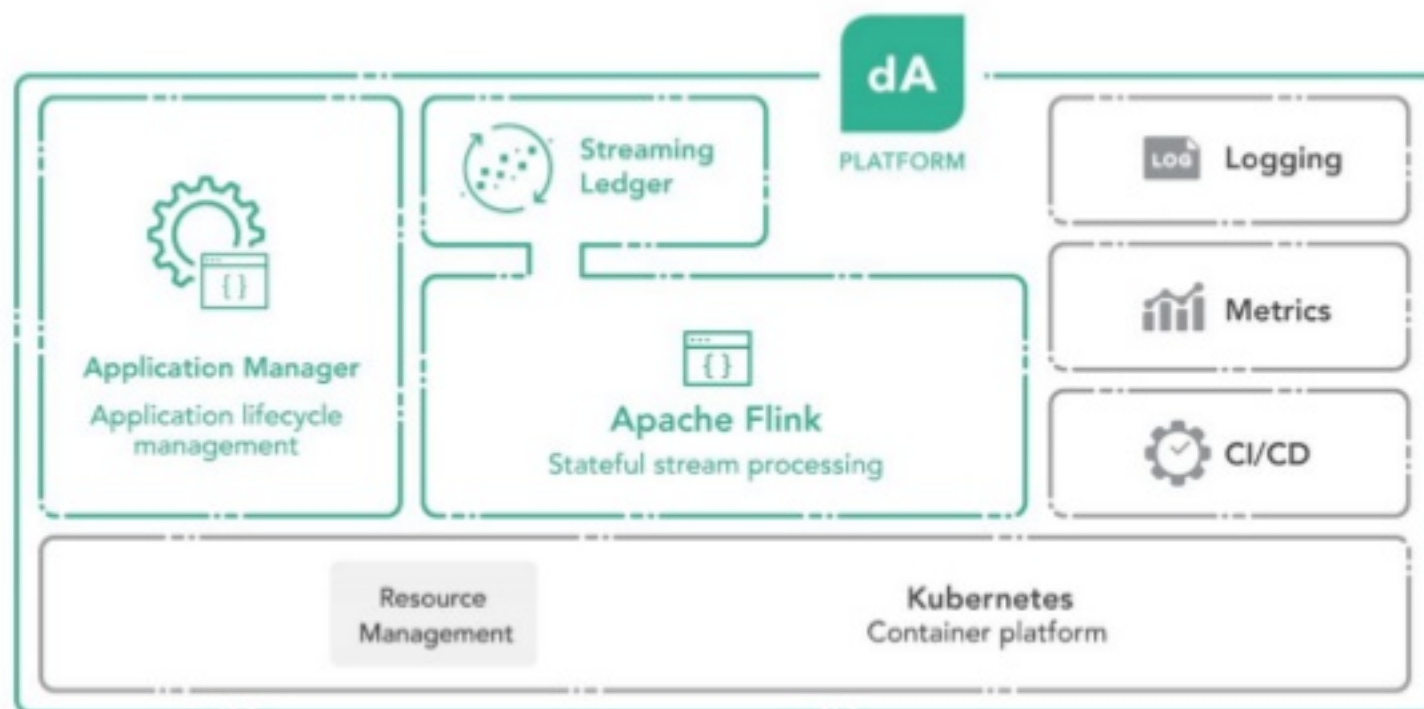
THANK YOU!

@igalshilman
@dataArtisans
@ApacheFlink

WE ARE HIRING
data-artisans.com/careers

dataArtisans