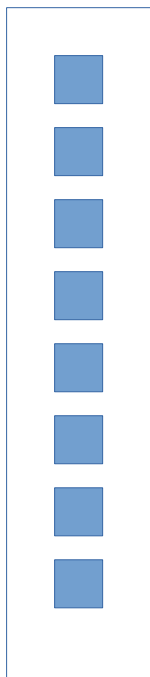


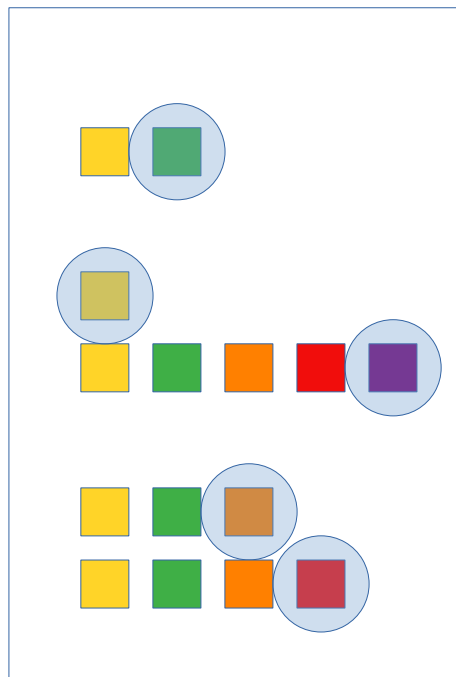
WHAT IS NEEDED

BEFORE

older_account

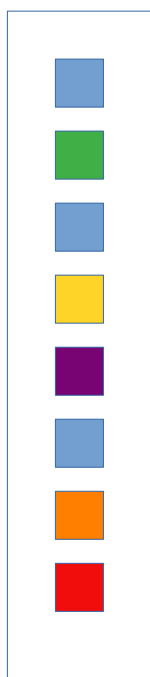


Account (day partition)



AFTER

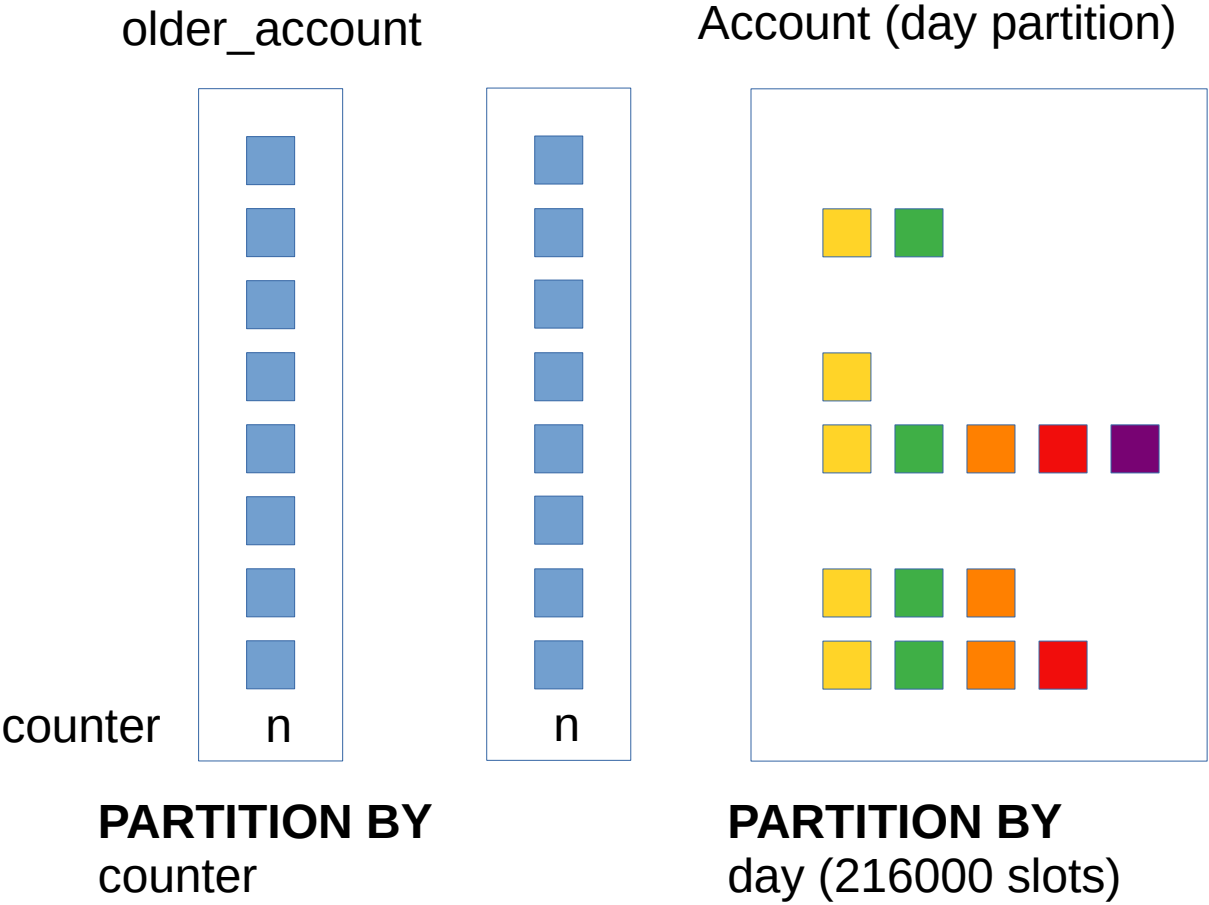
older_account



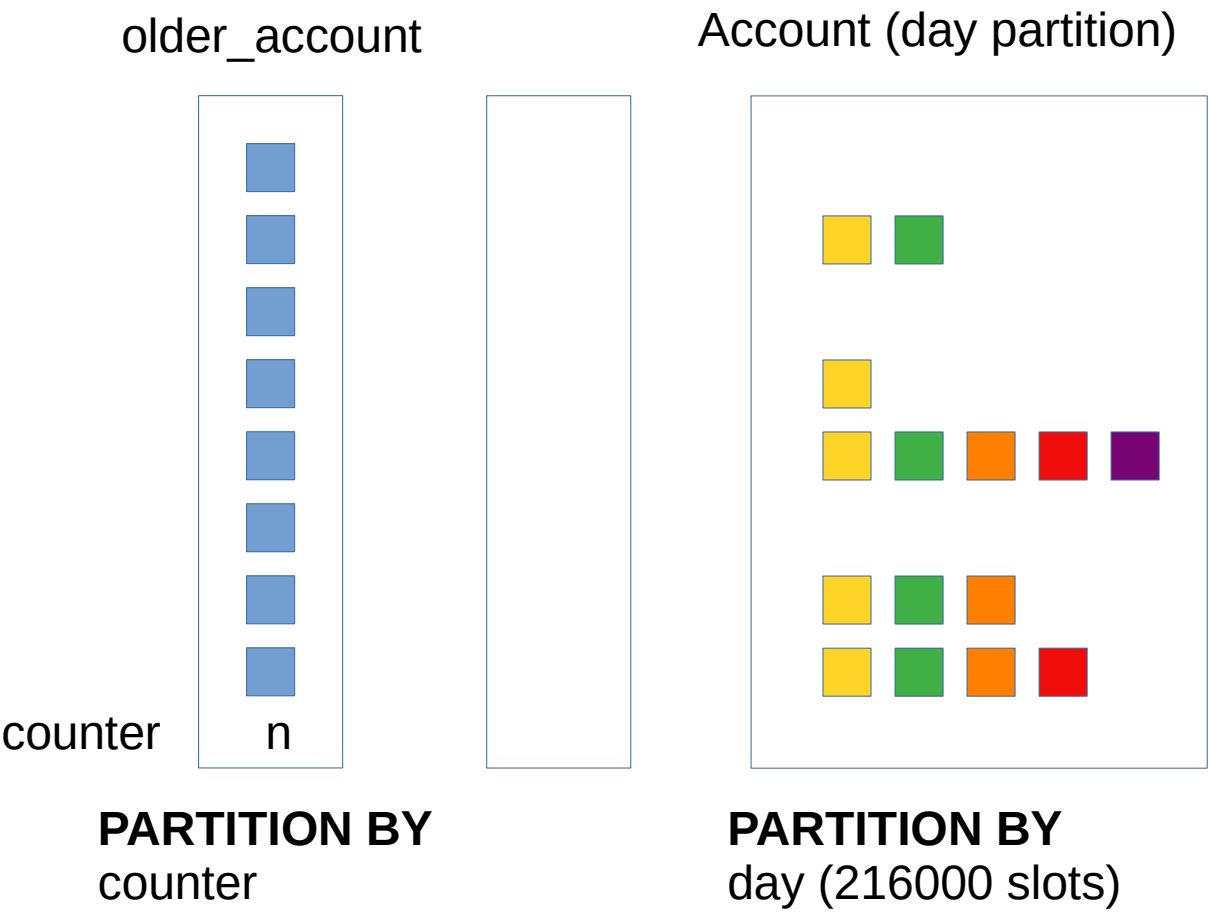
Account (day partition)



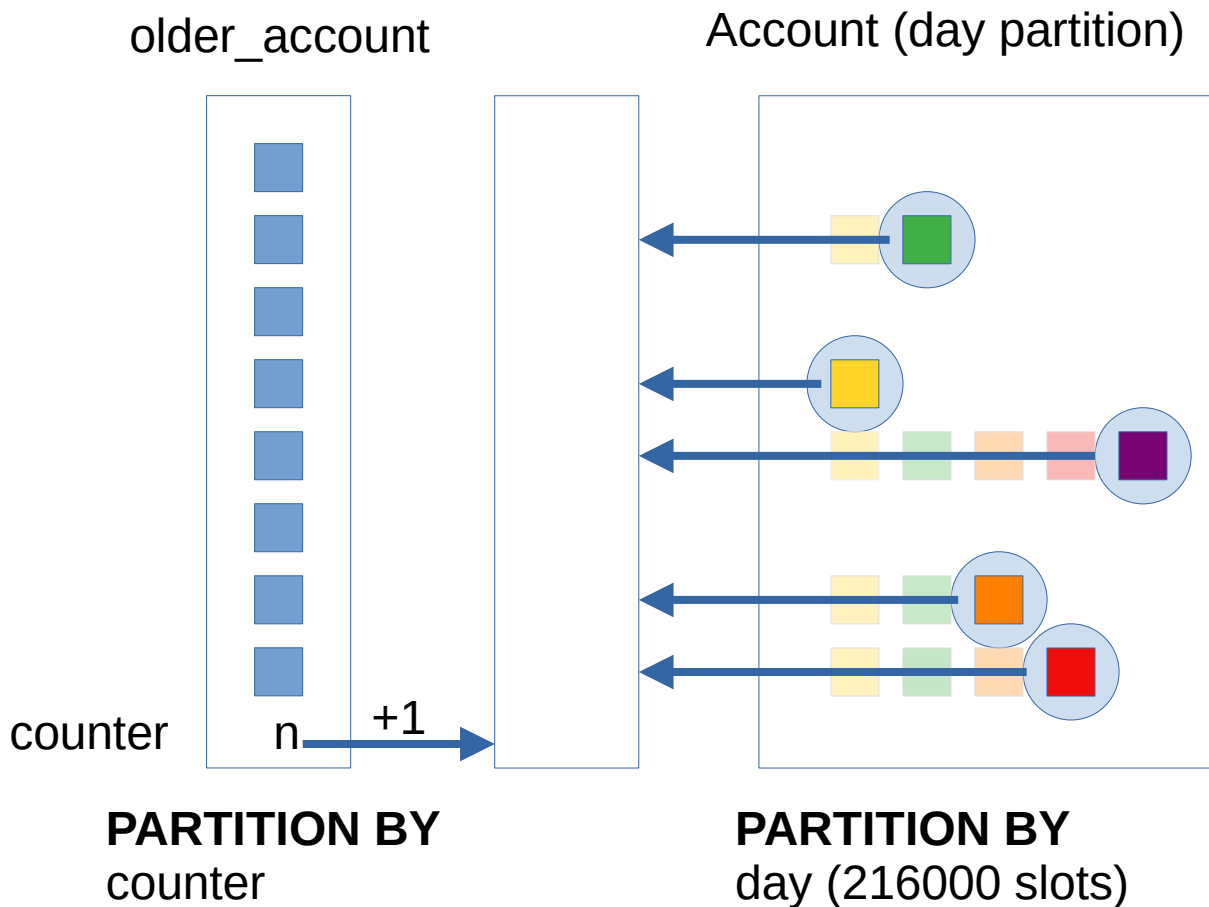
STEP 0: START



STEP 1: TRUNCATE INTERMEDIATE TABLE



STEP 2: SELECT FROM OUTDATED PARTITION

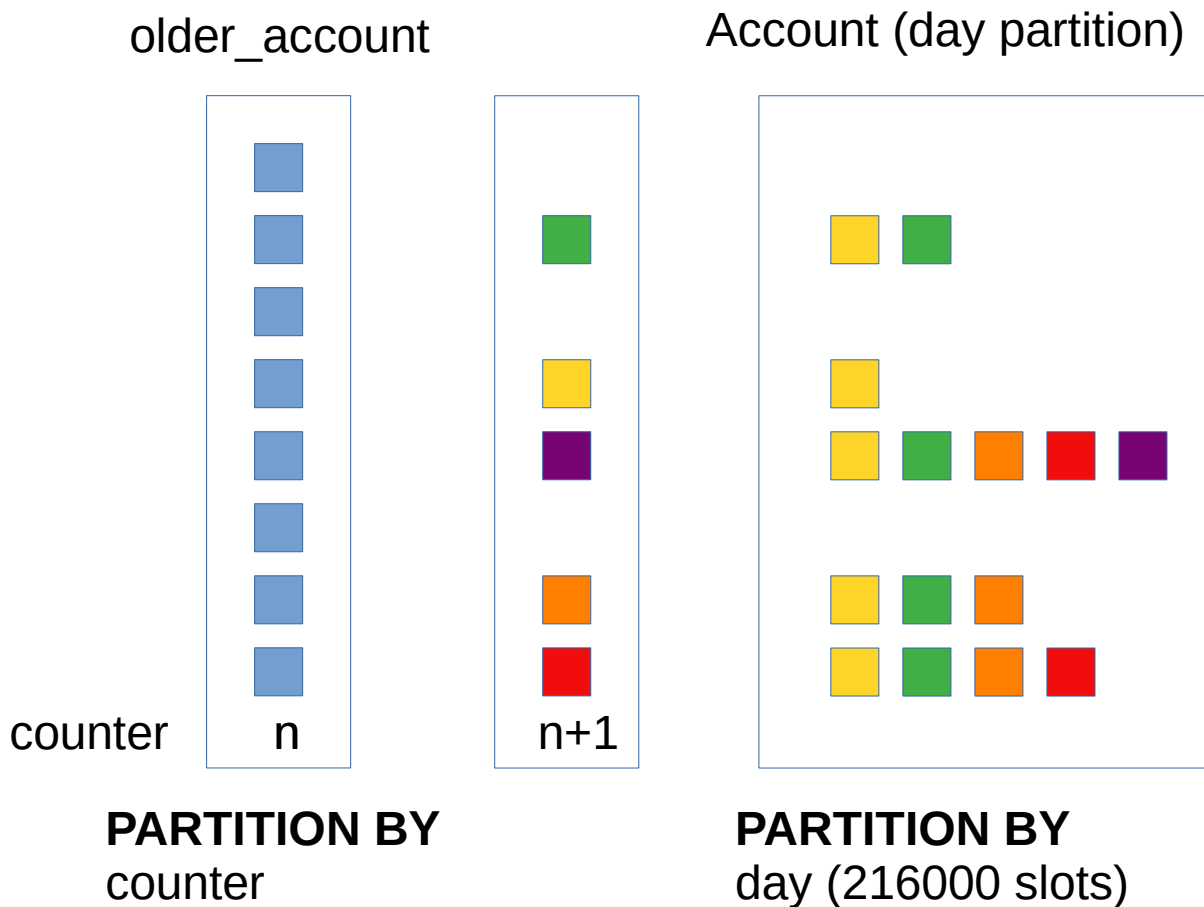


```

INSERT INTO events.items_to_retention
SELECT DISTINCT ON (ual.pubkey)
  ual.pubkey,
  ual.lamports,
  ual.owner,
  ual.executable,
  ual.rent_epoch,
  ual.`data`,
  ual.write_version,
  ual.txn_signature,
  ual.slot,
  ual.is_startup,
  ual.retrieved_time,
  (SELECT MAX(oal.retention_counter) + 1 FROM events.older_account_local oal)
FROM events.update_account_local ual
WHERE
  ual.slot > (SELECT MAX(oal.slot) FROM events.older_account_distributed oal)
  AND ual.slot <= (SELECT MAX(usd.slot) - 648000 FROM events.update_slot_distributed usd)
ORDER BY ual.pubkey DESC, ual.slot DESC, ual.write_version DESC;

```

STEP 2: SELECT FROM OUTDATED PARTITION

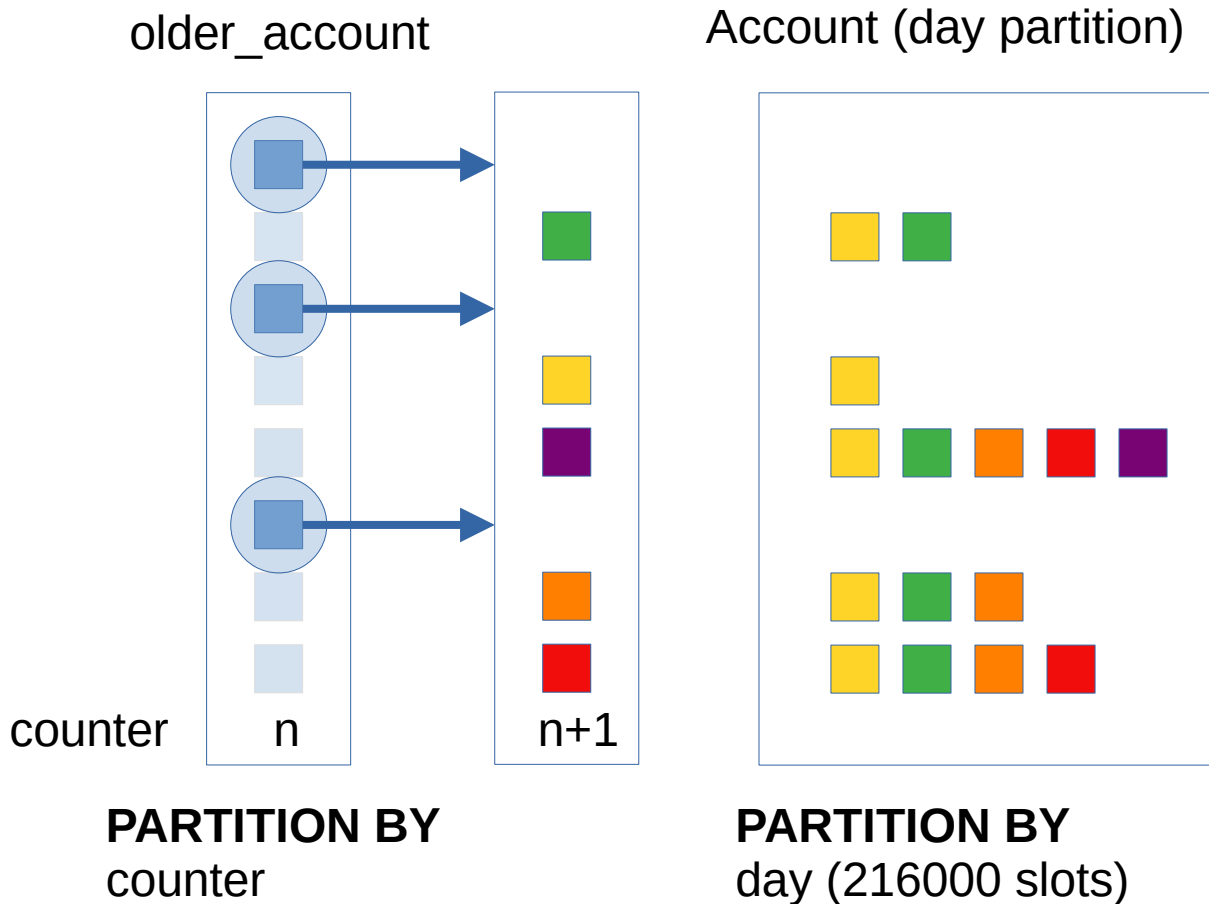


```

INSERT INTO events.items_to_retention
SELECT DISTINCT ON (ual.pubkey)
  ual.pubkey,
  ual.lamports,
  ual.owner,
  ual.executable,
  ual.rent_epoch,
  ual.`data`,
  ual.write_version,
  ual.txn_signature,
  ual.slot,
  ual.is_startup,
  ual.retrieved_time,
  (SELECT MAX(oal.retention_counter) + 1 FROM events.older_account_local oal)
FROM events.update_account_local ual
WHERE
  ual.slot > (SELECT MAX(oal.slot) FROM events.older_account_distributed oal)
  AND ual.slot <= (SELECT MAX(usd.slot) - 648000 FROM events.update_slot_distributed usd)
ORDER BY ual.pubkey DESC, ual.slot DESC, ual.write_version DESC;

```

STEP 3: COMPLETE FROM OLDER STATE

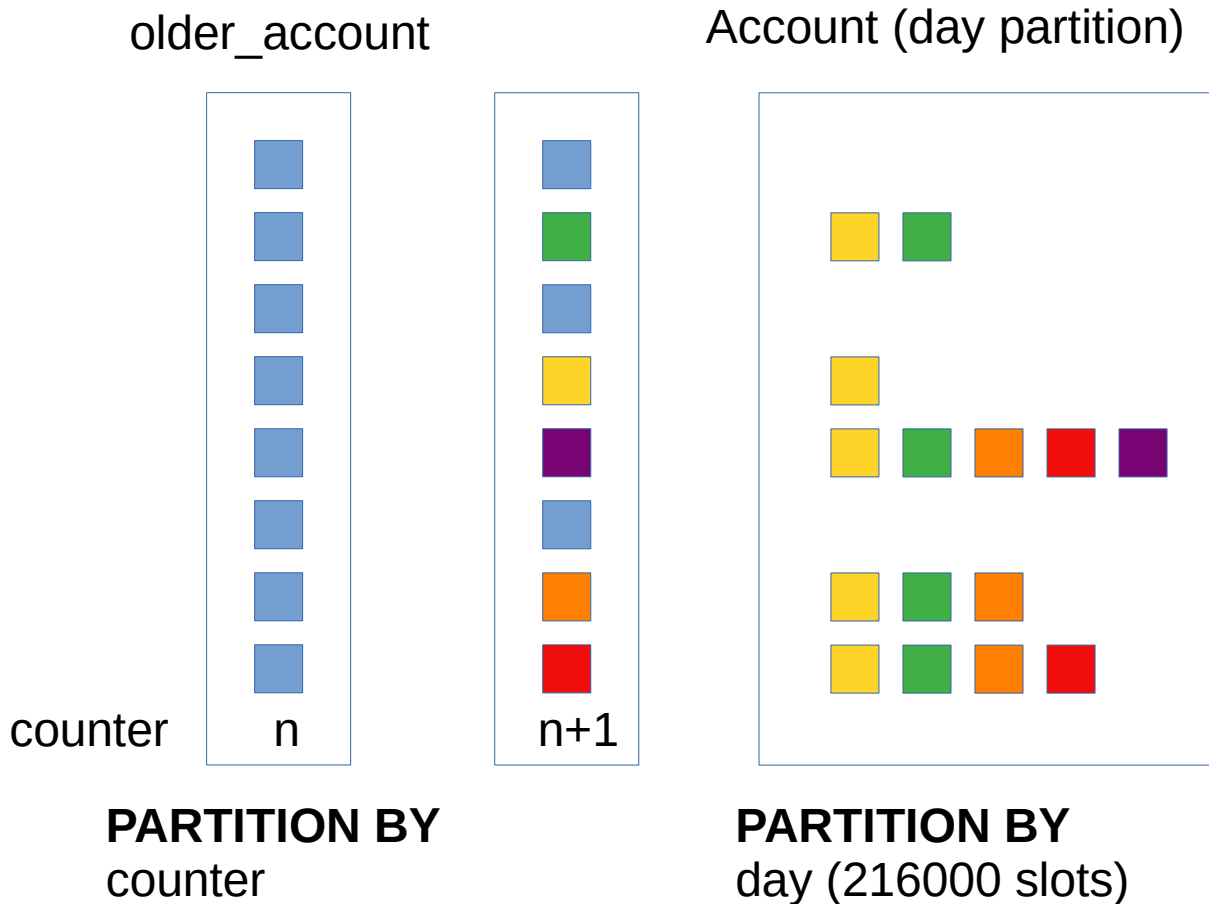


```

INSERT INTO events.items_to_retention
SELECT DISTINCT ON (oal.pubkey)
  oal.pubkey,
  oal.lamports,
  oal.owner,
  oal.executable,
  oal.rent_epoch,
  oal.`data`,
  oal.write_version,
  oal.txn_signature,
  oal.slot,
  oal.is_startup,
  oal.retrieved_time,
  (SELECT MAX(oal.retention_counter) + 1 FROM events.older_account_local oal)
FROM events.older_account_local oal
WHERE
  oal.pubkey NOT IN (SELECT pubkey FROM events.items_to_retention itr)
ORDER BY oal.pubkey DESC, oal.slot DESC, oal.write_version DESC;

```

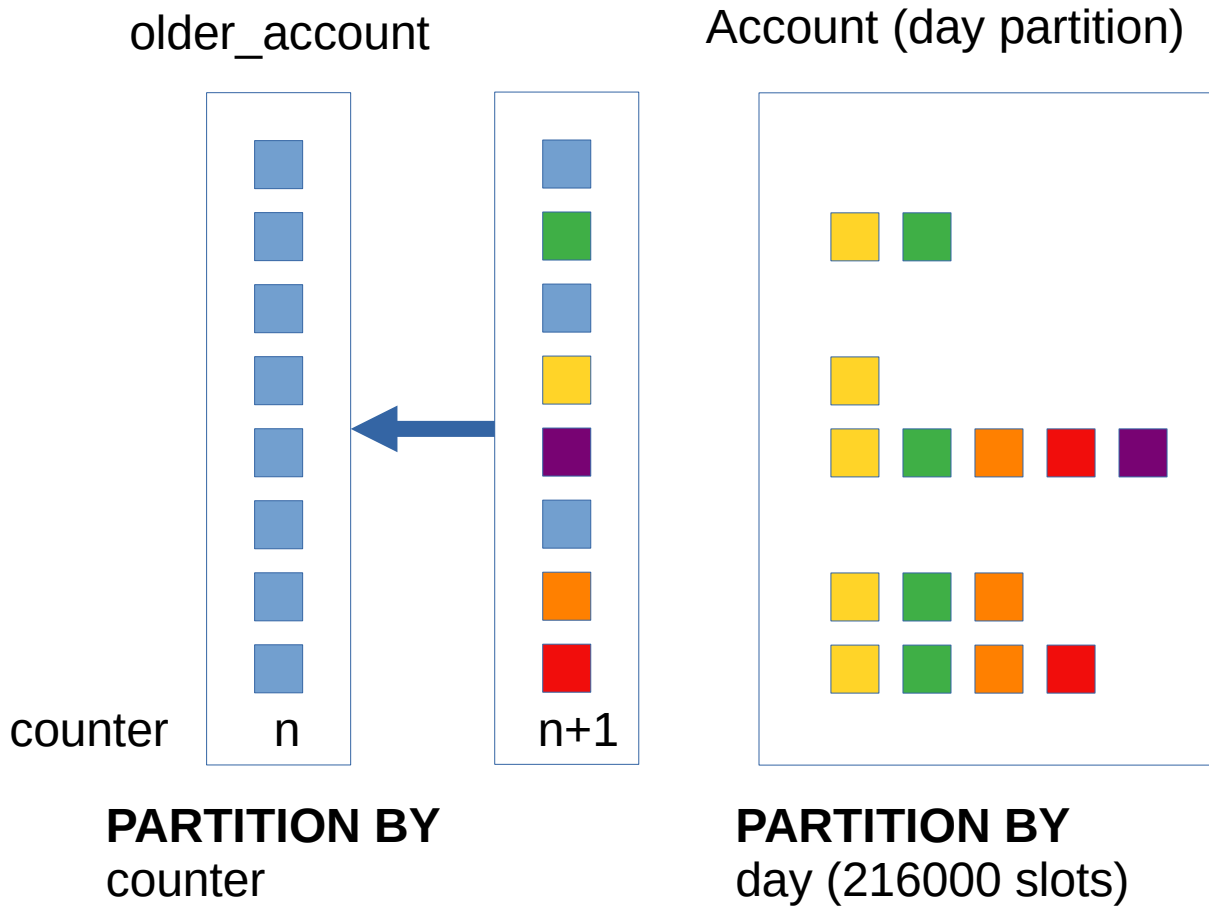
STEP 3: COMPLETE FROM OLDER STATE



```

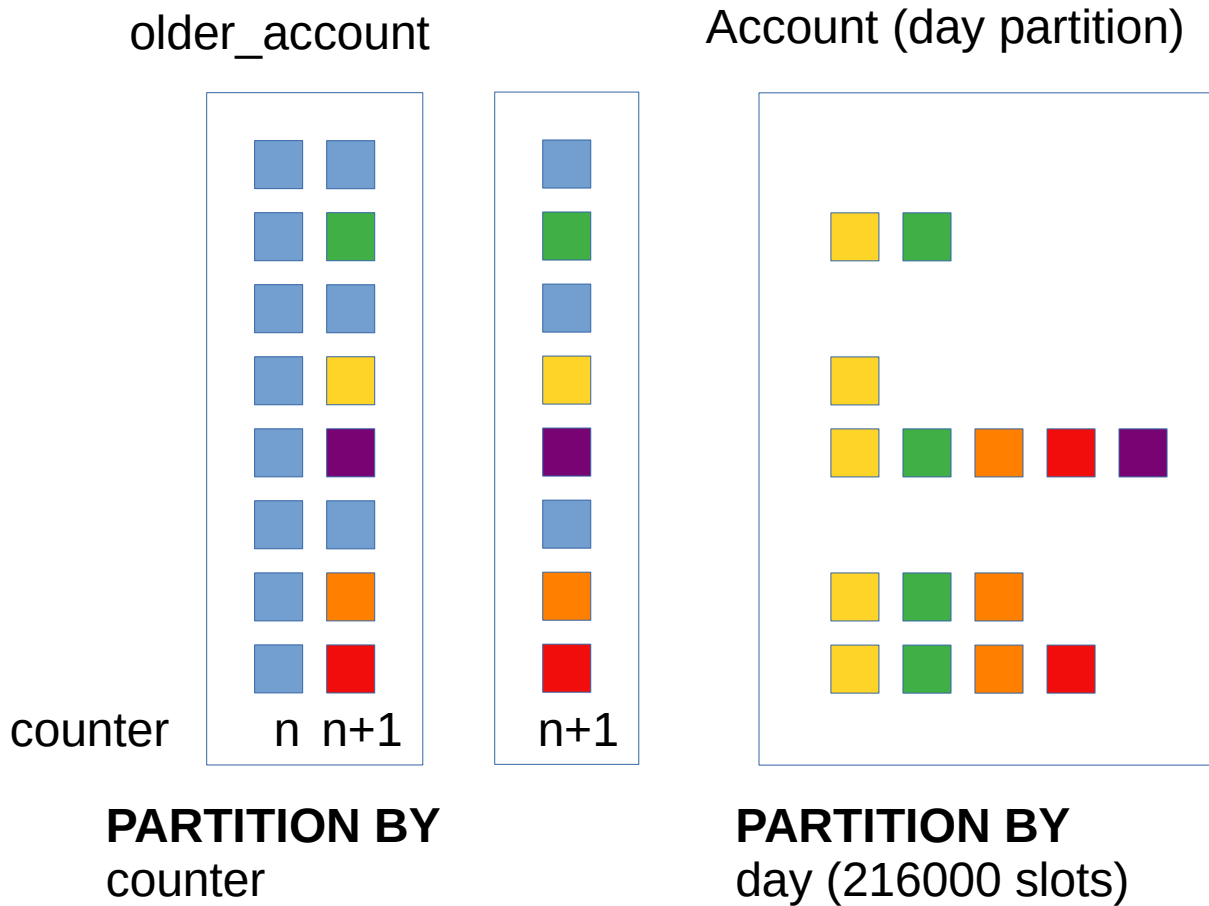
INSERT INTO events.items_to_retention
SELECT DISTINCT ON (oal.pubkey)
  oal.pubkey,
  oal.lamports,
  oal.owner,
  oal.executable,
  oal.rent_epoch,
  oal.`data`,
  oal.write_version,
  oal.txn_signature,
  oal.slot,
  oal.is_startup,
  oal.retrieved_time,
  (SELECT MAX(oal.retention_counter) + 1 FROM events.older_account_local oal)
FROM events.older_account_local oal
WHERE
  oal.pubkey NOT IN (SELECT pubkey FROM events.items_to_retention itr)
ORDER BY oal.pubkey DESC, oal.slot DESC, oal.write_version DESC;
  
```

STEP 4: INSERT



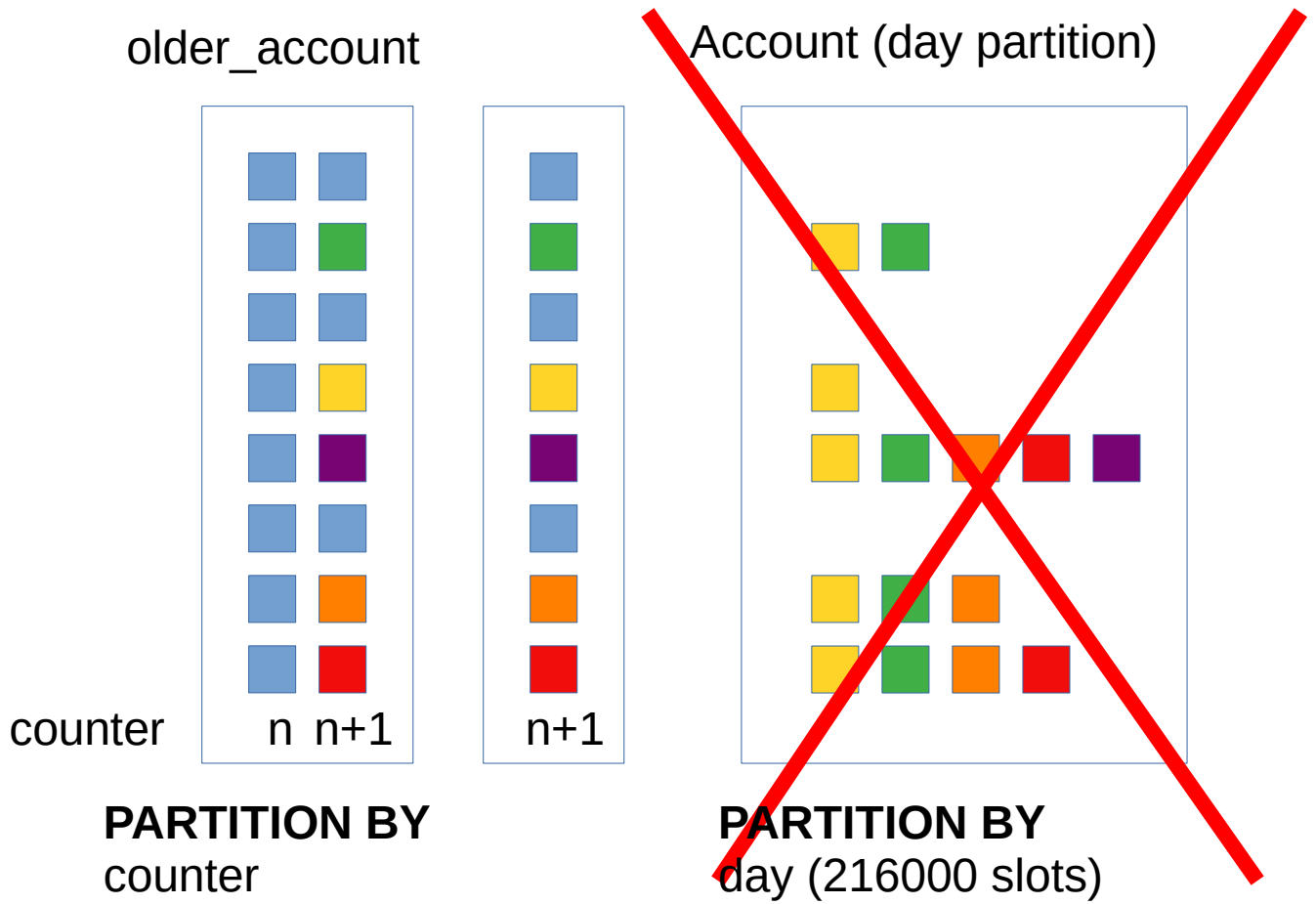
```
INSERT INTO events.older_account_local  
SELECT * FROM events.items_to_retention;
```


STEP 4: INSERT

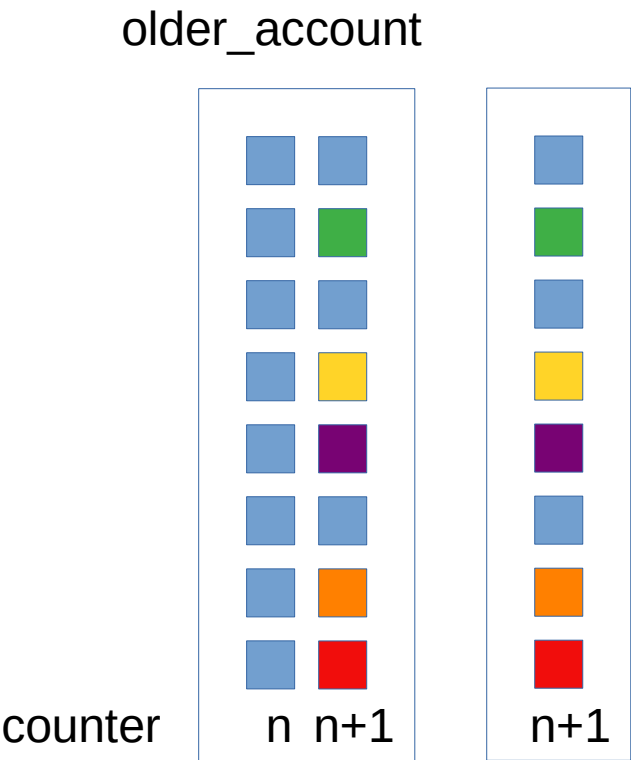


```
INSERT INTO events.older_account_local  
SELECT * FROM events.items_to_retention;
```

STEP 5: DELETE OUTDATED PARTITION

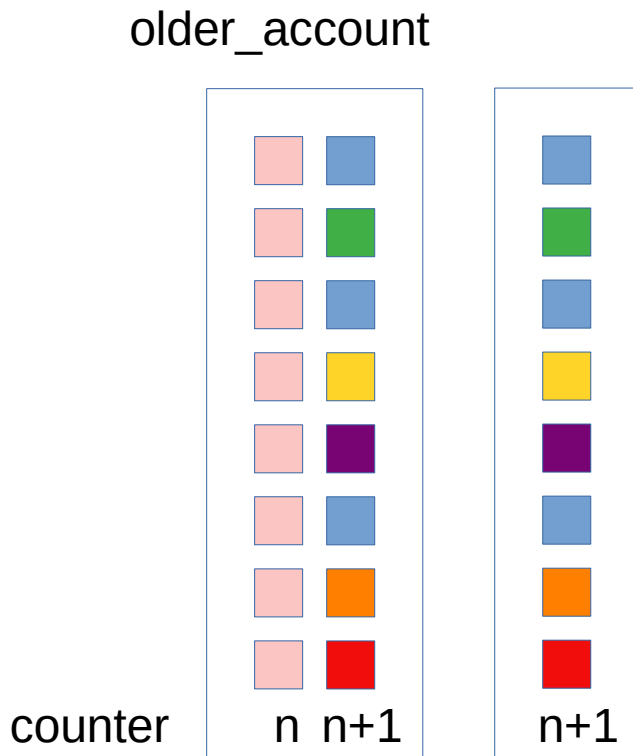


STEP 5: DELETE OUTDATED PARTITION



PARTITION BY
counter

STEP 6: DELETE PREV STATES

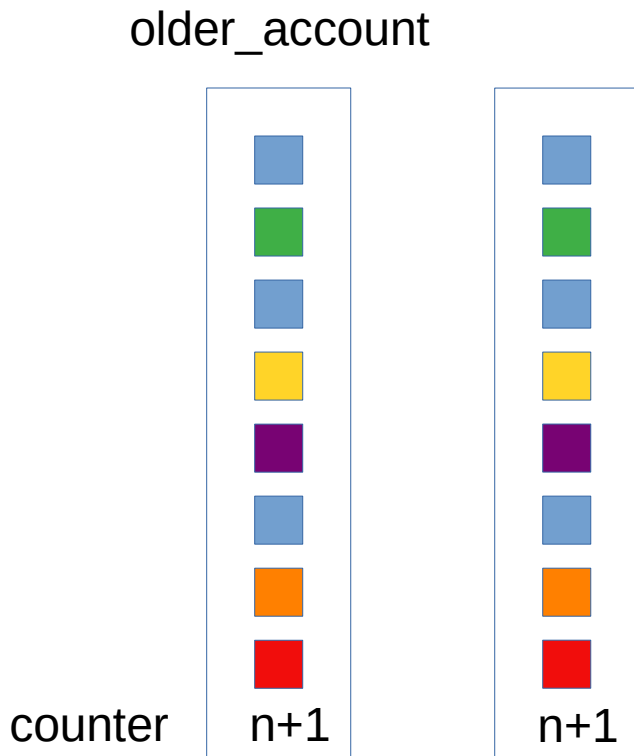


PARTITION BY
counter

```
CREATE FUNCTION IF NOT EXISTS get_older_account_part_to_drop ON CLUSTER 'events' AS ()->(
  SELECT MAX(retention_counter) - 1
  FROM events.older_account_local
  LIMIT 1
);

ALTER TABLE events.older_account_local DROP PARTITION tuple(get_older_account_part_to_drop())
```

STEP 6: DELETE PREV STATES



PARTITION BY counter

```
CREATE FUNCTION IF NOT EXISTS get_older_account_part_to_drop ON CLUSTER 'events' AS ()->(
  SELECT MAX(retention_counter) - 1
  FROM events.older_account_local
  LIMIT 1
);

ALTER TABLE events.older_account_local DROP PARTITION tuple(get_older_account_part_to_drop())
```