Report: Identity Value Incrementation on Failed Inserts in SQL Server

1. Introduction

In SQL Server, identity columns (IDENTITY(1,1)) automatically generate sequential values for primary keys. However, a common issue arises when an INSERT operation fails due to constraint violations—identity values still increment, leading to gaps in the sequence. This report examines the problem, its implications, and real-world examples without focusing on solutions.

2. Problem Statement

When an INSERT fails due to constraint violations (e.g., PRIMARY KEY, FOREIGN KEY, CHECK, NOT NULL), SQL Server still increments the identity counter. This results in unused identity values, creating gaps in the sequence.

Key Characteristics:

Automatic Increment Before Validation: The identity value is assigned before constraint checks. Transaction-Independent: Even if a transaction rolls back, the identity value is not reused. Permanent Gaps: Once skipped, identity values are never reclaimed.

3. Why This Is Problematic

While gaps in identity values are generally harmless, they can cause issues in certain scenarios:

User Expectations

Users may expect strictly sequential IDs (e.g., invoice numbers, order IDs).

Gaps can lead to confusion ("Why is Order #5 missing?").

Audit & Compliance

Some industries require unbroken sequences for tracking (e.g., financial transactions).

Gaps may raise red flags during audits.

Poor User Experience

Applications displaying IDs may appear unreliable if gaps are visible.

Customers may perceive missing numbers as errors.

Wasted Values

In systems with limited integer ranges (e.g., INT), skipped values reduce available IDs

4. Real-World Examples

Example 1: Failed CHECK Constraint

```
create table dept
(
 did int primary key,
 dname varchar(20)
)
create table emp
(
 eid int identity(1,1),
 ename varchar(20),
 eadd varchar(20) default 'cairo',
 hiredate date default getdate(),
 BD date,
 age as(year(getdate())-year(BD)), -- ربخ والتاريخ يتغير
 sal int,
 overtime int,
 --we will not give them data type
 --لكت بتحصله موجود ك ستركتشر بس هو غلى الهاردسك دريفن
 netsal as(isnull(sal,0)+isnull(overtime,0)) persisted,--
 gender varchar(1),
 hour_rate int not null,
 Dnum int,
 constraint c1 primary key(eid,ename),
 constraint c2 unique(sal),
 constraint c3 unique(overtime),
 constraint c4 check(sal>1000),
 constraint c5 check(eadd in('alex','mansoura','cairo')),
 constraint c6 check(overtime between 100 and 500),
 constraint c7 check(gender='m' or gender='F'),
 constraint c8 foreign key(Dnum) references Dept(did)
           on delete set null on update cascade
```

Example 2: FOREIGN KEY Violation

```sql
INSERT INTO dept (did, dname) VALUES
(1, 'Human Resources'),
(2, 'IT'),
(3, 'Finance'),
(4, 'Marketing'),
(5, 'Operations');

INSERT INTO emp (ename, eadd, BD, sal, overtime, gender, hour_rate, Dnum) VALUES
('John Smith', 'cairo', '1985-03-15', 2000, 200, 'm', 25, 1),
('Sarah Johnson', 'alex', '1990-07-22', 1500, 150, 'F', 20, 2),
('Michael Brown', 'mansoura', '1988-11-05', 1800, 300, 'm', 22, 3),
('Emily Davis', 'cairo', '1992-05-30', 2200, 400, 'F', 28, 4),
('Robert Wilson', 'alex', '1983-09-12', 2500, 250, 'm', 30, 5),
('Jennifer Lee', 'mansoura', '1995-02-18', 1200, 350, 'F', 18, 1),
('David Miller', 'cairo', '1987-12-10', 3000, 500, 'm', 35, 2),
('Jessica Taylor', 'alex', '1993-06-25', 1700, 100, 'F', 21, 3),
('Daniel Anderson', 'mansoura', '1980-08-08', 2800, 450, 'm', 32, 4),
('Lisa Martinez', 'cairo', '1991-04-14', 1900, 340, 'F', 24, 5);
```

Example 3: NOT NULL Constraint Violation

```sql
INSERT INTO emp (ename, eadd, BD, sal, overtime, gender, hour_rate, Dnum) VALUES
('Jon Smith', 'cairo', '1985-03-15', 2300, 390, 'm', 25, 1);
INSERT INTO emp (ename, eadd, BD, sal, overtime, gender, hour_rate, Dnum) VALUES
('Sara', 'alex', '1990-07-22', 600, 180, 'F', 20, 2);
INSERT INTO emp (ename, eadd, BD, sal, overtime, gender, hour_rate, Dnum) VALUES
('Mich', 'mansoura', '1988-11-05', 1860, 330, 'm', 22, 3);
```

Result: `Eid 37` is skipped permanently.

eid	ename	eadd	hiredate	BD	age	sal	overtime	netsal	gender	hour_rate	Dnum
11	John Smith	cairo	2025-05-19	1985-03-15	40	2000	200	2200	m	25	1
12	Sarah Johnson	alex	2025-05-19	1990-07-22	35	1500	150	1650	F	20	2
13	Michael Brown	mansoura	2025-05-19	1988-11-05	37	1800	300	2100	m	22	3
14	Emily Davis	cairo	2025-05-19	1992-05-30	33	2200	400	2600	F	28	4
15	Robert Wilson	alex	2025-05-19	1983-09-12	42	2500	250	2750	m	30	5
16	Jennifer Lee	mansoura	2025-05-19	1995-02-18	30	1200	350	1550	F	18	1
17	David Miller	cairo	2025-05-19	1987-12-10	38	3000	500	3500	m	35	2
18	Jessica Taylor	alex	2025-05-19	1993-06-25	32	1700	100	1800	F	21	3
19	Daniel Anderson	mansoura	2025-05-19	1980-08-08	45	2800	450	3250	m	32	4
20	Lisa Martinez	cairo	2025-05-19	1991-04-14	34	1900	340	2240	F	24	5
36	Jon Smith	cairo	2025-05-19	1985-03-15	40	2300	390	2690	m	25	1
38	Mich	mansoura	2025-05-19	1988-11-05	37	1860	330	2190	m	22	3
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

5. Implications

Scenario	Consequence
User-facing IDs	Confusion over missing numbers
Audit trails	May appear incomplete
Limited ID ranges	Wasted values (e.g., `INT` maxes at ~2B)
Reporting	Queries expecting sequential IDs may fail

6. Conclusion

SQL Server's behavior of incrementing identity values on failed inserts is by design, ensuring uniqueness rather than continuity. While gaps are usually harmless, they can cause issues in systems requiring strict sequencing. Understanding this behavior helps in designing robust databases and setting proper user expectations.

KEY TAKEAWAYS FROM THIS ANALYSIS

1. **UNIQUENESS OVER CONTINUITY**
   SQL Server prioritizes guaranteed uniqueness of identity values over maintaining an unbroken sequence. This tradeoff was consciously made to ensure database reliability under all conditions.
2. **IRREVERSIBLE VALUE CONSUMPTION**
   Once an identity value is generated—whether the insert succeeds or fails—that value is permanently consumed and cannot be reclaimed, even if the transaction rolls back.
3. **SYSTEM-LEVEL VS BUSINESS-LEVEL IDENTIFIERS**
   This behavior highlights why identity columns should typically serve as internal system identifiers rather than customer-facing business keys where sequence continuity might be expected.
4. **ARCHITECTURAL IMPLICATIONS**
   Applications requiring gap-free sequences must implement alternative designs rather than relying on native identity column behavior.