ORACLE

Oracle Database 21c

BlockChain Tables

Witold Świerzy

EMEA Data Domain Expert

Blockchain Enables & Optimizes Business & Organizational Networks

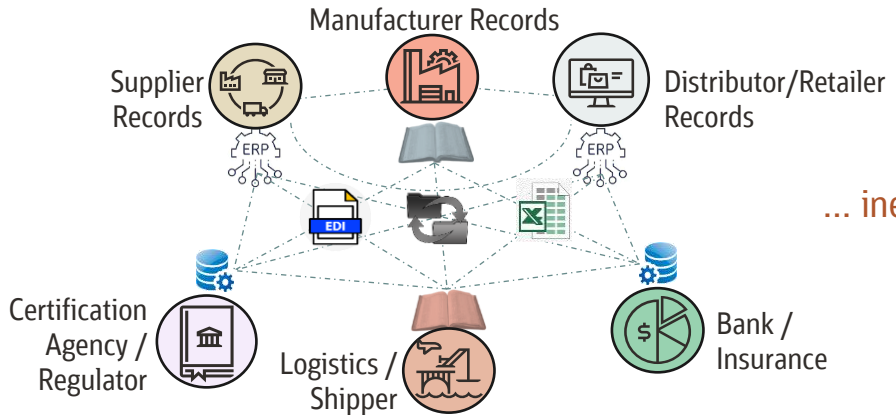
Blockchain replaces complex point-to-point data flows & removes intermediaries with a *distributed ledger* that creates a shared source of truth & trusted transactions.

Common challenges & situations it can address:

- **High reconciliation cost** and operations impact of **siloed** data across divisional or company boundaries
- **Lack of verifiability, risk of human errors or fraud** among your partners or ecosystem participants
- **Cost, risk, and delays from intermediaries** the business operations depends on
- **Lack of real-time visibility** among suppliers, distributors, or customers due to batch data updates
- **Poor traceability** or lack of audit trail for regulatory compliance or internal best practices
- **Inability to track** physical or digital assets effectively and leverage their value in a tokenized economy

Fractured Data Siloes in Many Enterprise Ecosystems

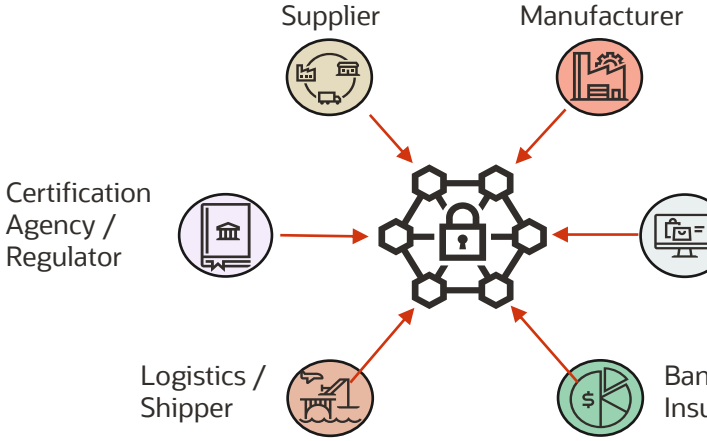
Point-to-point data flow via spreadsheets, EDI, or B2B file transfer



... inefficient, expensive, vulnerable

Enterprise blockchain replaces vulnerable data sharing with

A trusted, Distributed Ledger with real-time updates, providing...



Permissioned Interoperable, Resilient, Scalable

...single source of truth with non-repudiation, consensus, traceability, immutability

Oracle's Vision for Enterprise Blockchain

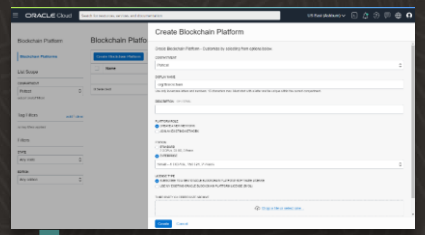
Blockchain can enable *rapid development* of **business & organizational networks** and *significant optimization* of existing networks via transparent, consensus-based trusted transactions maintaining single source of truth between independent parties without the need for intermediaries.

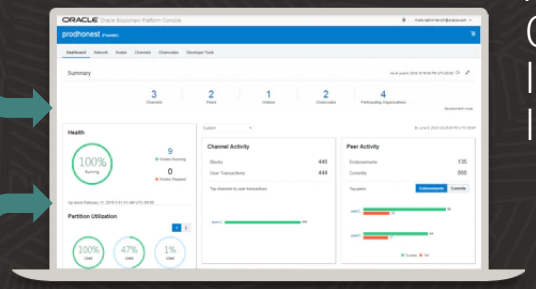
Market Strategy and Offerings

- Make blockchain adoption *easy* and *quick* for enterprises
- Provide market-leading *cloud* and *on-premises* offerings for both customers and developers who want to build their own solutions – **Oracle Blockchain Platform** based on *Hyperledger Fabric*.
- Embed blockchain in enterprise-ready solutions for business users – SaaS Applications – e.g., **Intelligent Track & Trace**.
- Leverage blockchain techniques in **Oracle Database's** Crypto-Secure Data Management features.

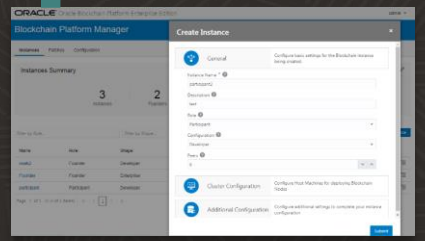
Oracle Blockchain Technology Overview

Cloud provisioning





Admin Console



On-prem provisioning

Blockchain Platform Cloud

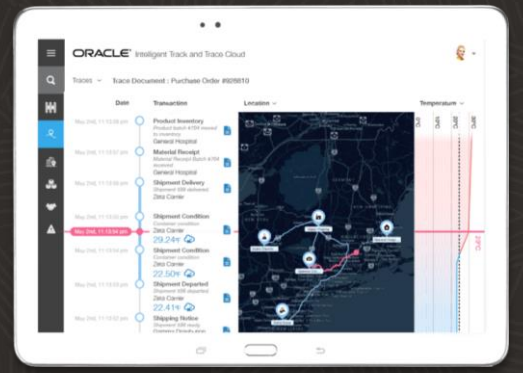
Managed Blockchain-as-a-Service based on Hyperledger Fabric
Highly Available, Resilient, Scalable
Build and deploy smart contracts in Go, Java, JavaScript
Manage Confidential Transactions
API Gateway & Bi-directional Events
Operations Tools/DevOps APIs
Integration with Oracle DB & Analytics
Interoperable, multi-cloud topology

Blockchain Platform EE (On-Prem)

Pre-assembled platform for on-prem
Simple install using virtualization
Same features & APIs as Cloud
Built on Docker containers
Same console Interface
Identity management using LDAP/AD

Intelligent Track and Trace

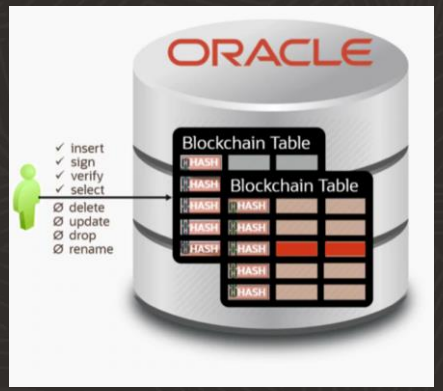
Prebuilt Business-ready SaaS
End-to-end supply chain visibility
Stakeholder on-boarding
Simulation Capability
Integrates data from *OTM Cloud, Mfg. Cloud, Procurement Cloud, Inventory Mgmt. Cloud, IoT Fleet Monitoring Cloud*
Oracle Integration Adapter for 3rd party and on-premises apps



Users can track, trace, and monitor transactions and their associated assets, items, and documents.

Database Blockchain Tables

Insert-only DB Tables
Cryptographic hash-chained rows
Tamper-proof & verifiable
Optional user signatures and DB-signed digest
Standard DB access: SQL, PL/SQL, JDBC, etc. and tools
Protect centralized ledgers against user or admin fraud



Blockchain Tables

TRADE LEDGER

ID	User	Value	Hash
1	Tom	500	ADSJS
2	Carol	176	%SHS
3	Steve	900	SH@1
4	[illegible]	[illegible]	DHD3
5	Mike	332	*EGG
6	Sarah	632	AH11
7	Eve	25	LIO$
8	Prisha	850	SHS4

BLOCKCHAIN TABLE

- Brings many of the benefits of blockchain technology to enterprise apps
- Insert-only tables where rows are cryptographically chained
  - Chain can be verified and signed by participants
- Simple to integrate into apps
  - Look like standard tables with declarative SQL
  - Full analytics and transactions on blockchain data

Blockchain Tables

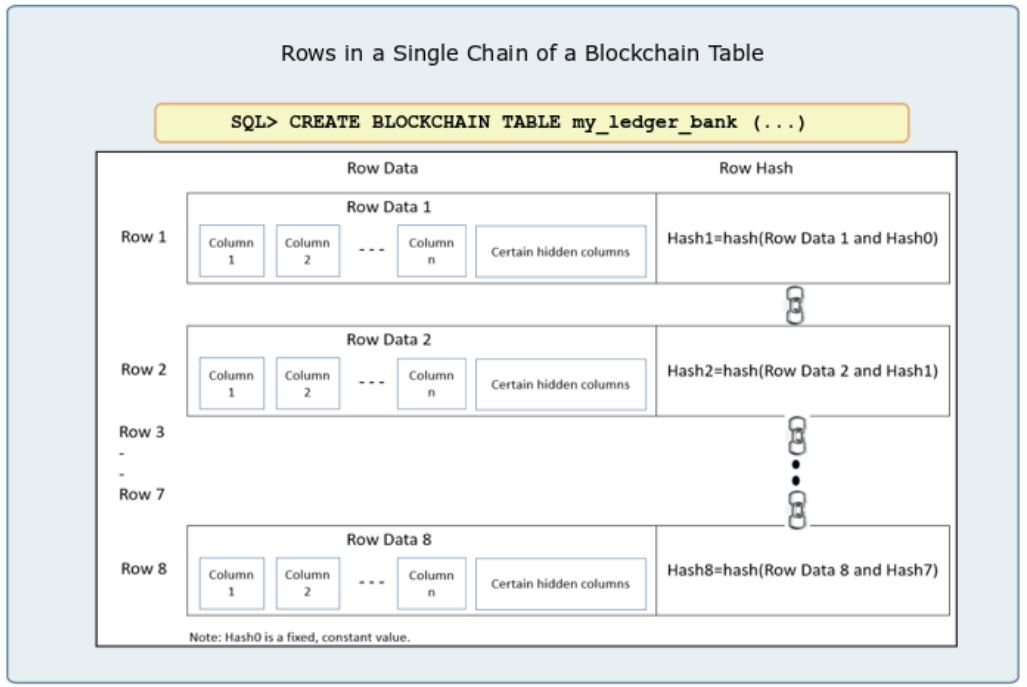
TRADE LEDGER

ID	User	Value	Hash
1	Tom	500	ADSJS
2	Carol	176	%SHS
3	Steve	900	SH@1
4	[illegible]	[illegible]	DHD3
5	Mike	332	*EGG
6	Sarah	632	AH11
7	Eve	25	LIO$
8	Prisha	850	SHS4

BLOCKCHAIN TABLE

- Rows in a blockchain table are made tamper-resistant by special sequencing and chaining algorithms
- Users can verify that rows have not been tampered
- A hash value that is part of the row metadata is used to chain and validate rows.

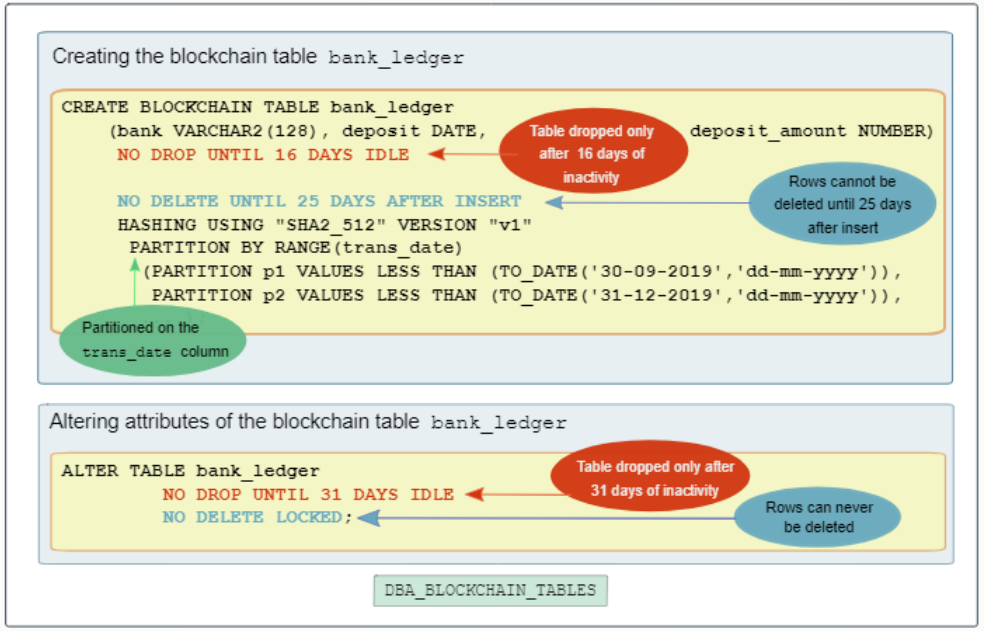
Blockchain Tables



- **Blockchain tables** are used to implement centralized blockchain applications where the central authority is the Oracle Database.

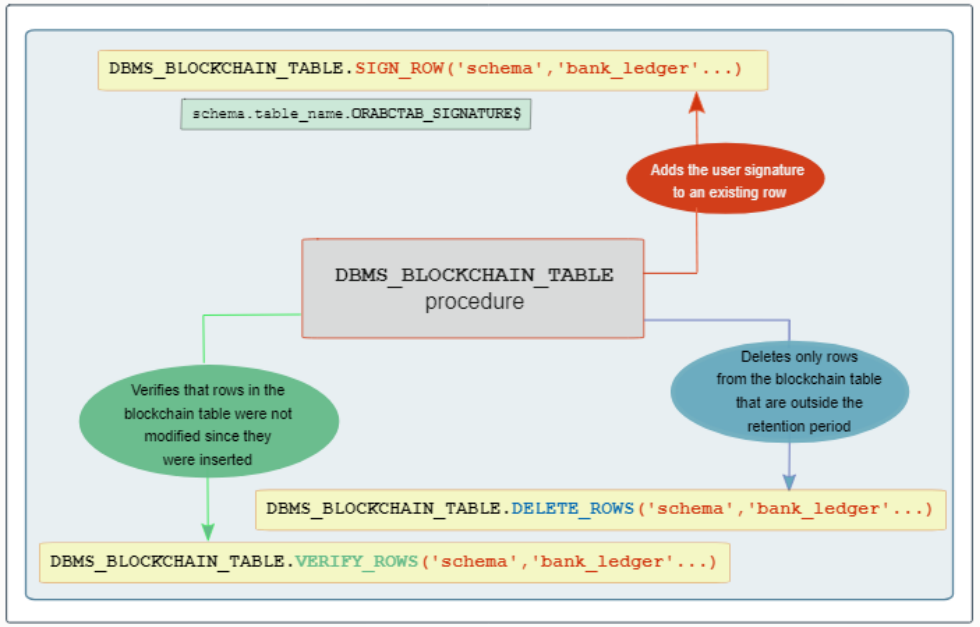
- **Rows in a blockchain table are tamper-proof.** Each row contains a cryptographic hash value which is based on the data in that row and the hash value of the previous row in the chain. If a row is tampered with, the hash value of the row changes and this causes the hash value of the next row in the chain to change. An optional user signature can be added to a row for enhanced fraud detection.

Blockchain Tables



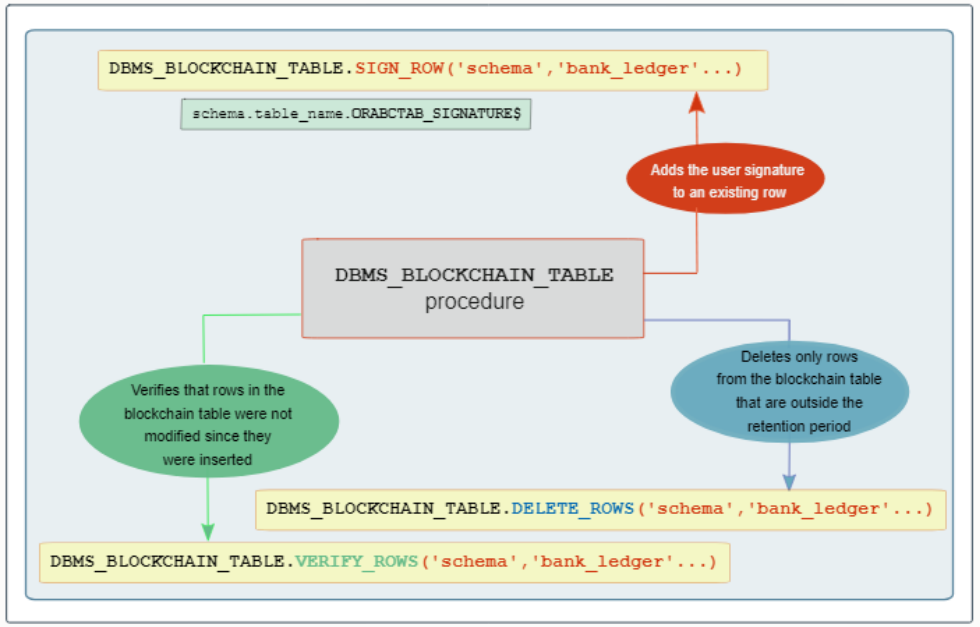
- **Blockchain tables can be indexed and partitioned.**
- It is possible to control **whether and when rows are deleted** from a blockchain table and whether the table can be dropped
- Blockchain tables can be used along with (regular) **tables in transactions and queries.**

Blockchain Tables



- **Signing** a row sets a user signature for a previously created row. A signature provides additional security against tampering.
- **Validation** is possible by using DBMS_BLOCKCHAIN_TABLE.VERIFY_ROWS procedure, which verifies that rows in a blockchain table were not modified since they were inserted.

Blockchain Tables



- Only rows that are outside the retention period can be **deleted** from a blockchain table. The **DBMS_BLOCKCHAIN_TABLE.DELETE_ROWS** procedure deletes all rows or rows that were created before a specified date

Blockchain Tables

```
CREATE BLOCKCHAIN TABLE ledger_emp (employee_id
NUMBER, salary NUMBER)
                    NO DROP UNTIL 31 DAYS IDLE
                    NO DELETE LOCKED
                    HASHING USING "SHA2_512"
VERSION "v1";
```

```
 SELECT row_retention, row_retention_locked,
                     table_inactivity_retention, hash_algorithm
               FROM   user_blockchain_tables
               WHERE  table_name='LEDGER_EMP';
```

- To create a block chain table there is need to use CREATE BLOCKCHAIN TABLE statement
- New data dictionary views: USER/ALL/DBA_BLOCKCHAIN_TABLES
- Every blockchain table contains set of hidden columns (they are not visible in DESCRIBE command results)

```
SQL> SELECT internal_column_id "Col ID", SUBSTR(column_name,1,30) "Column Name", SUBSTR(data_type,1,30) "Data Type",
data_length "Data Length" FROM user_tab_cols WHERE table_name = 'LEDGER_EMP' ORDER BY internal_column_id;
Col ID     Column Name Data         Type                           Data Length
---------- ------------------------ ------------------------------ -----------
...
         3 ORABCTAB_INST_ID$ NUMBER 22
         4 ORABCTAB_CHAIN_ID$ NUMBER 22
         5 ORABCTAB_SEQ_NUM$ NUMBER 22
         6 ORABCTAB_CREATION_TIME$ TIMESTAMP(6) WITH TIME ZONE 13
         7 ORABCTAB_USER_NUMBER$ NUMBER 22
         8 ORABCTAB_HASH$ RAW 2000
         9 ORABCTAB_SIGNATURE$ RAW 2000
        10 ORABCTAB_SIGNATURE_ALG$ NUMBER 22 11 ORABCTAB_SIGNATURE_CERT$ RAW 16 12 ORABCTAB_SPARE$ RAW 2000
```



ORACLE