

ORACLE®

# **Bob Thome**

## Group Product Manager

# Oracle Streams

# Agenda

- Information Sharing
- Oracle Streams
- Deployments for specific markets
- Comparison of Oracle9i Information Sharing features
- Q&A

# Information Sharing

- What is information Sharing?
- The moving or copying of information from users, applications, and databases who have it, to users, applications, and databases who need it.
- Examples...

# Information Sharing Examples

- **Investment Bank** replicates data worldwide so branches can operate autonomously
- **Brokerage** combines database changes with wire service messages and performs analysis
- **Trading Company** captures DML from the database and broadcasts to customers
- **Online Retailer** replicates data to another database to provide a failover database

# Information Sharing Examples

- **Auction House** replicates auction data to offload auction searches and reporting
- **B2B Exchange** captures transactions from business partners and processes them through its CRM, ERP, and legacy apps
- **International Bank** retains and routes customer orders to various stock exchanges
- **Manufacturer** extracts changed records from financial applications to load a DW

# Features for Information Sharing

- **Features Introduced In Prior Releases**
  - **Advanced Queuing** -- Message Queuing
  - **Advanced Replication** -- Replication of Data
  - **Synchronous Change Data Capture** -- DW Loading
- **New in Oracle9i Database Release 2**
  - **Oracle Streams** -- a comprehensive information sharing feature and infrastructure

# Oracle Streams

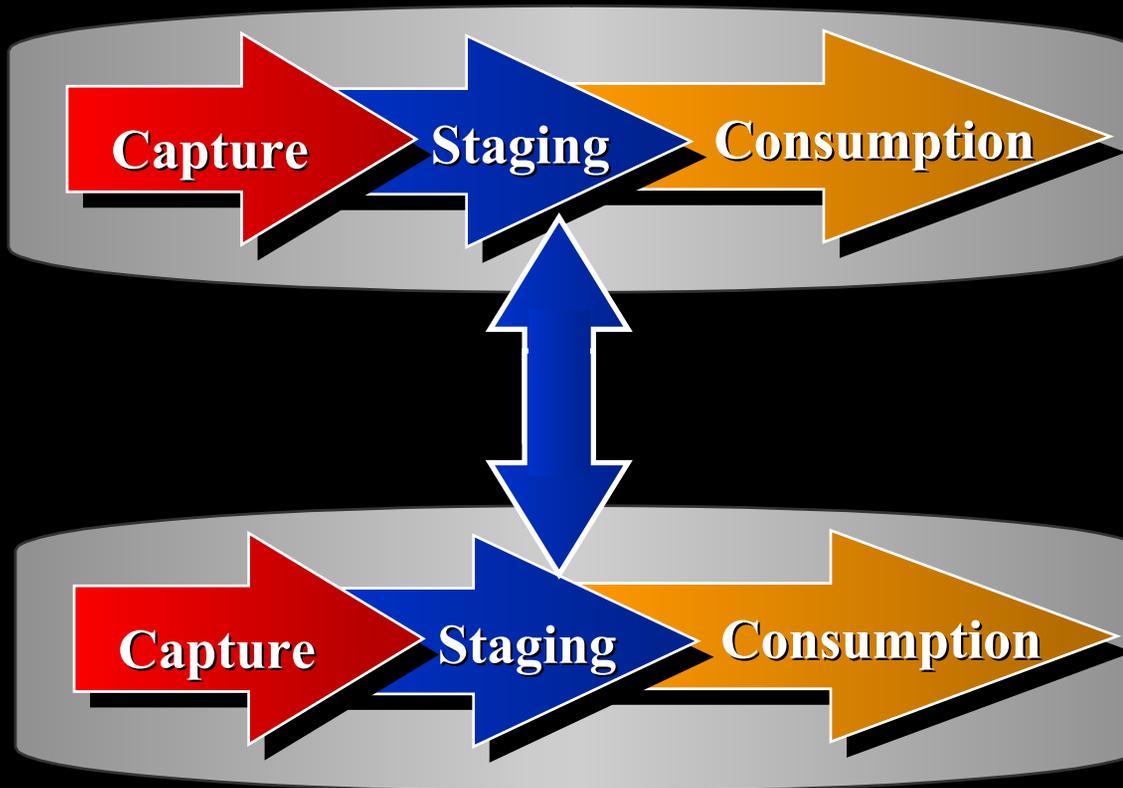
- A new solution for information sharing
- Provides a unified architecture for all information sharing solutions
  - uniquely flexible replication
  - message queuing
  - data warehouse loading
  - event management and notification

# Streams Basic Elements



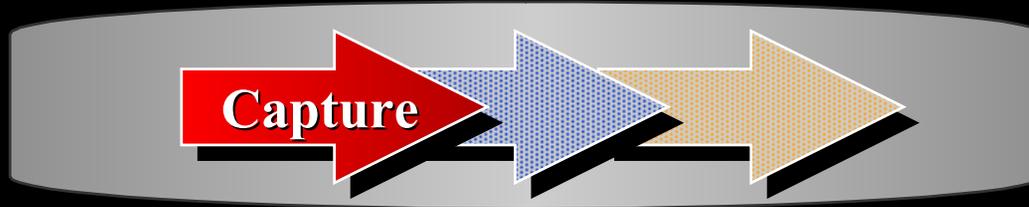
- Three basic elements in each database
  - Capture
  - Staging
  - Consumption (apply)

# Multi-Database Streams



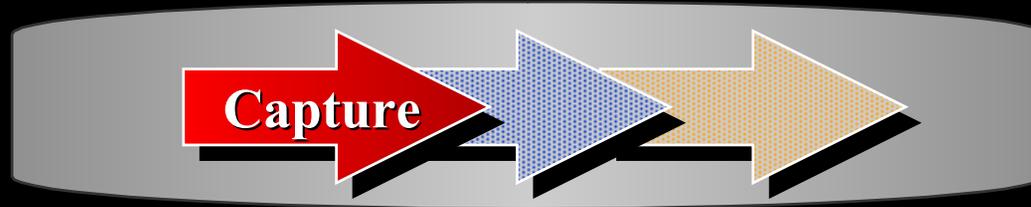
- A stream can contain multiple elements from multiple databases
- Events flow between staging areas

# Capture



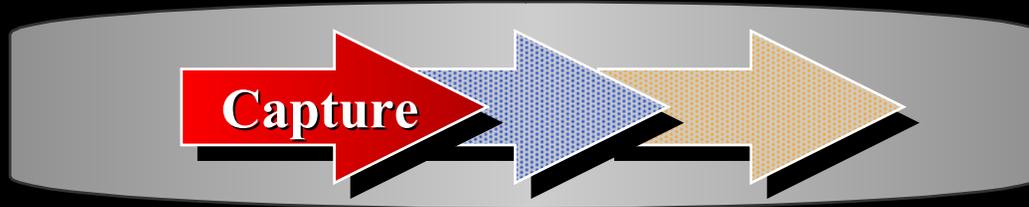
- Streams captures events
  - Implicitly: log-based capture of DML and DDL
  - Explicitly: Direct enqueue of user messages
- Captured events are published in the staging area

# Log-Based Change Capture



- Low overhead, low latency change capture
  - Changes to the database are written to the online redo log
  - Oracle Streams can extract changes from the log as it is written (mining the active log)
  - Changes are formatted as a Logical Change Record (LCR), a representation of the operation, and the before and after images

# Direct Enqueue



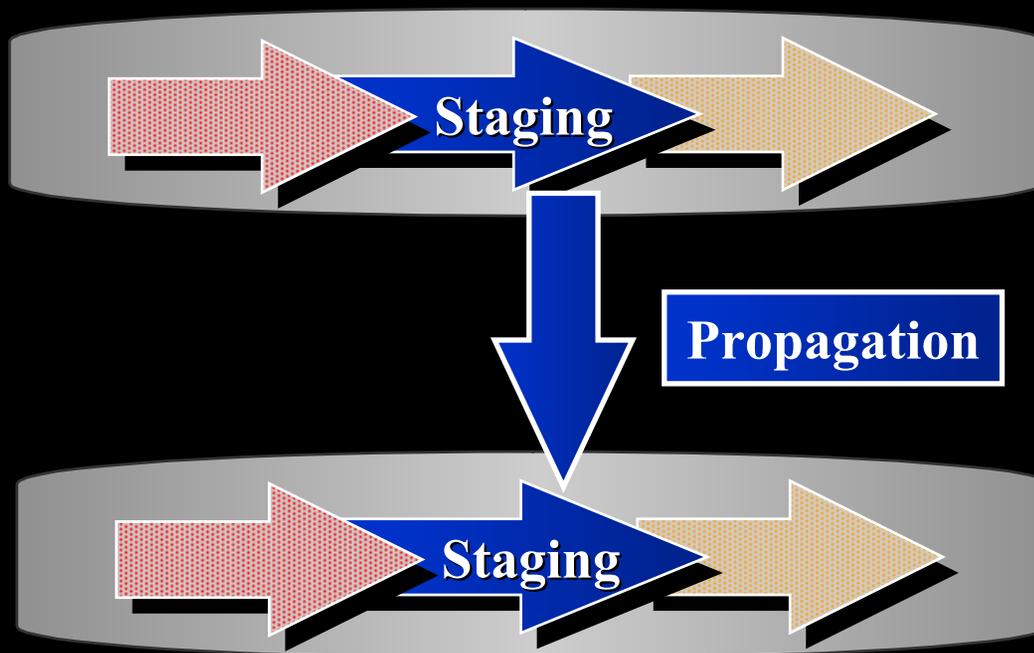
- User applications can explicitly enqueue user messages into the staging area
  - Multiple open interfaces supported: JMS, C, PLSQL, SOAP (XML/HTTP)
  - Allows applications to communicate at a higher level
  - Allows users to introduce events into the stream from non-Oracle systems

# Staging



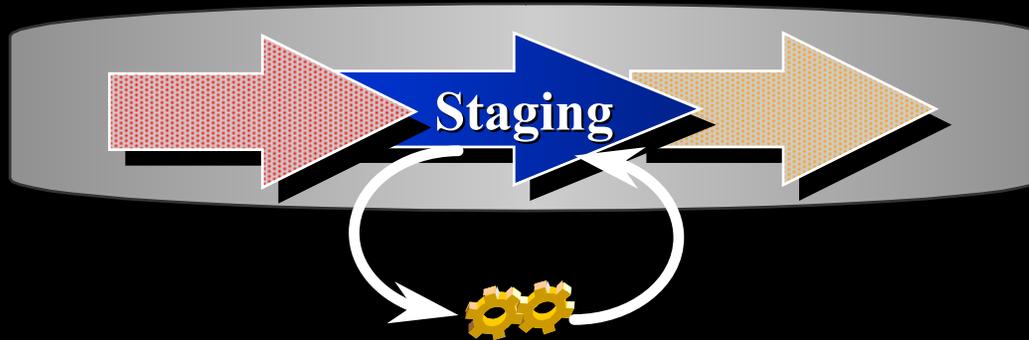
- Streams publishes captured events into a staging area
  - Implemented as a queue
  - Supports for new self-describing type “any” datatype allows a single staging area to hold any type of data
  - All events, LCRs and user-messages, can be staged in the same queue
  - Messages remain in staging area until consumed by all subscribers

# Staging Area Propagation



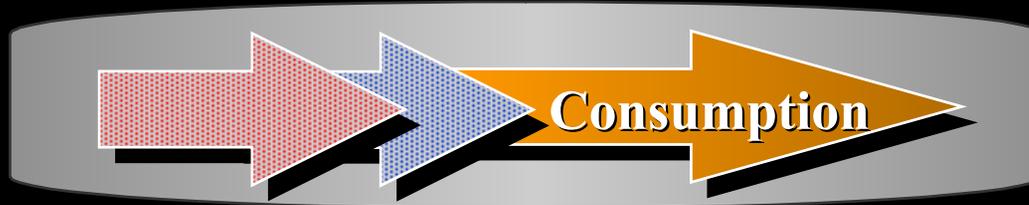
- Other staging areas can subscribe to events
  - in same database
  - in a remote database
- Events can be routed through a series of staging areas

# Transformations



- Transformations can be performed
  - as events enter the staging area
  - as events leave the staging area
  - as events propagate between staging areas
- Transformation examples
  - change format, data type, column name, table name

# Consumption



- Staged events are consumed by subscribers
  - Implicitly: Apply Process
    - Default Apply
    - User-Defined Apply
  - Explicitly: Application dequeue via open interfaces
    - JMS, C, PLSQL, SOAP (XML/HTTP)

# Default Apply



- The default apply engine will directly apply the DML or DDL represented in the LCR
  - apply to local Oracle table
  - apply via DB Link to non-Oracle table
- Automatic conflict detection with optional resolution
  - unresolved conflicts placed in exception queue
- Parallel apply maximizes concurrency

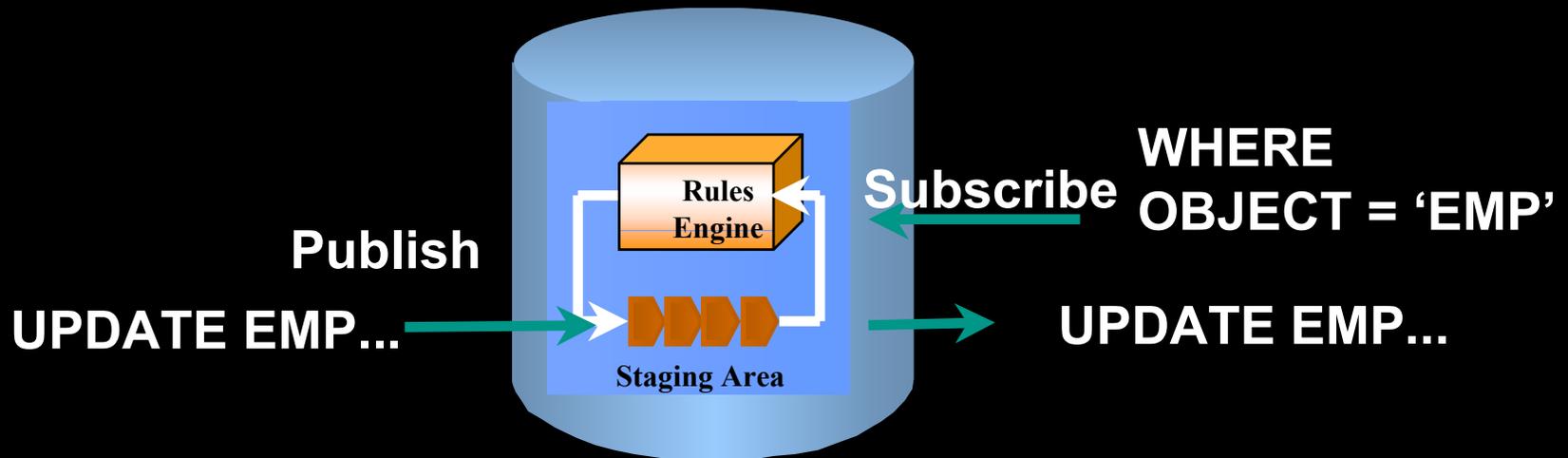
# User-defined Apply



- User-written custom apply functions
- Written in PL/SQL, Java, C, C++
- Uses:
  - custom transformations
  - column subsetting
  - normalizing or denormalizing data
  - populating related fields or tables

# Rule-based Subscription

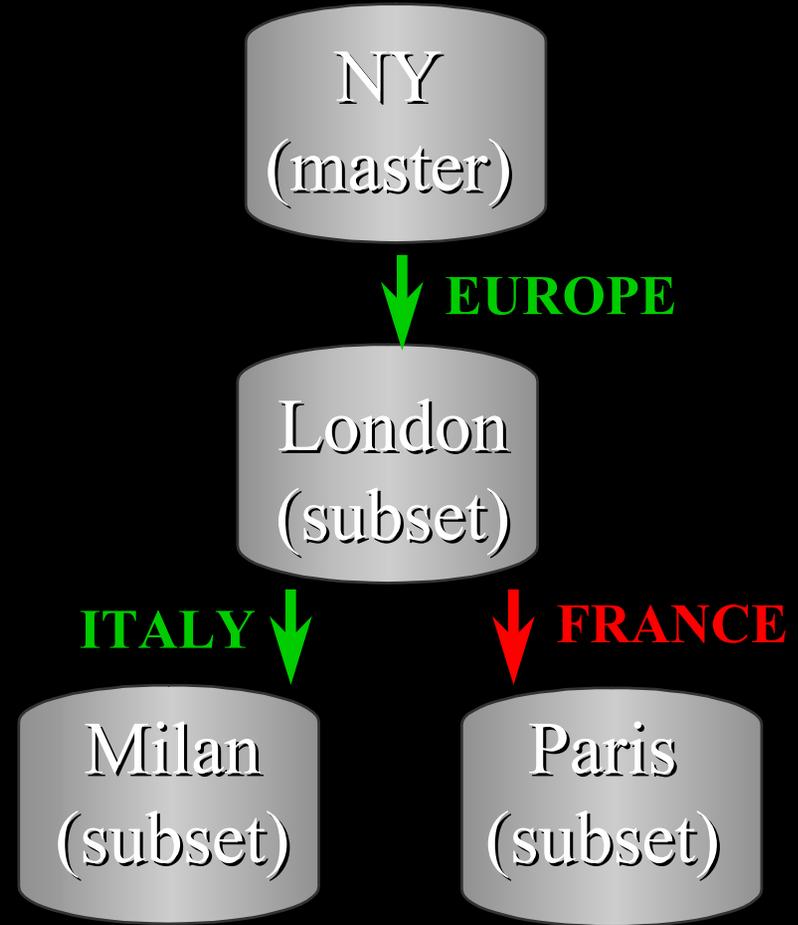
- Consumers subscribe to published events
- Content-based subscriptions limit delivered events to those meeting the subscription criteria
- Rules govern capture, staging, and consumption



# Directed Networks

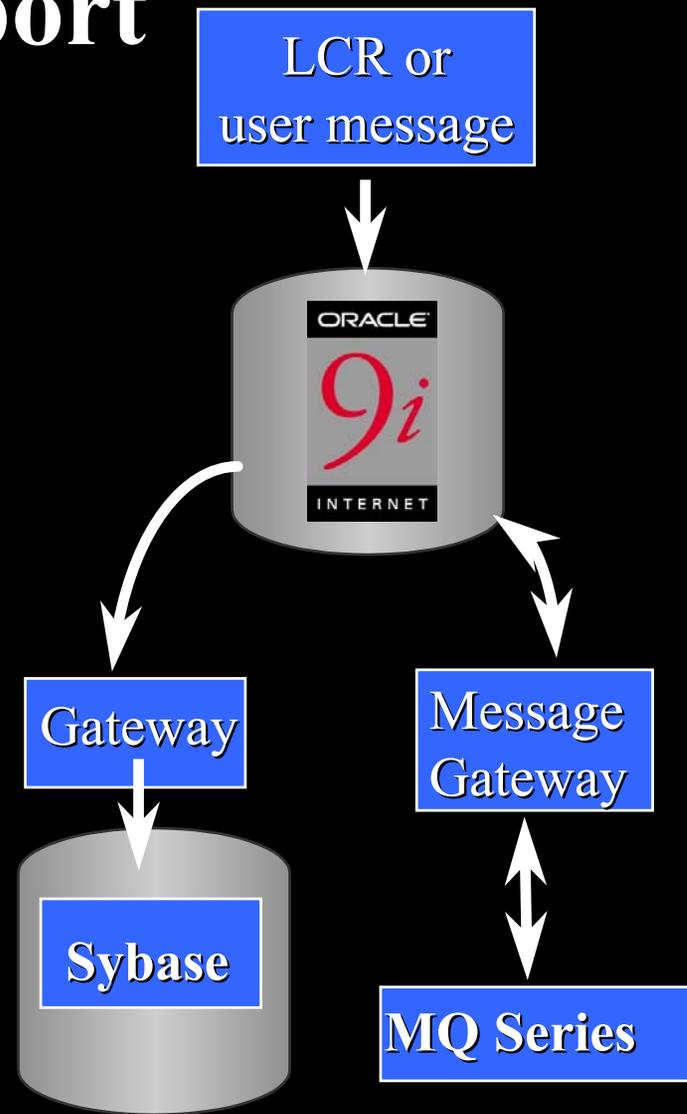
- Propagation independent of Apply
- Rules-based subscription determine if event is locally applied
  - London applies UK only
- WAN Friendly
  - Send once, fan out
  - NY-->London, London-->Milan, London-->Paris

INSERT ...  
VALUES ('EUROPE', 'ITALY')



# Heterogeneous Support

- Oracle to non-Oracle apply via gateway
  - Apply process on Oracle node applies change
- Non-Oracle to Oracle change capture supported via explicit enqueue of LCRs
- Message Gateways
  - MQ Series



# Streams Deployments

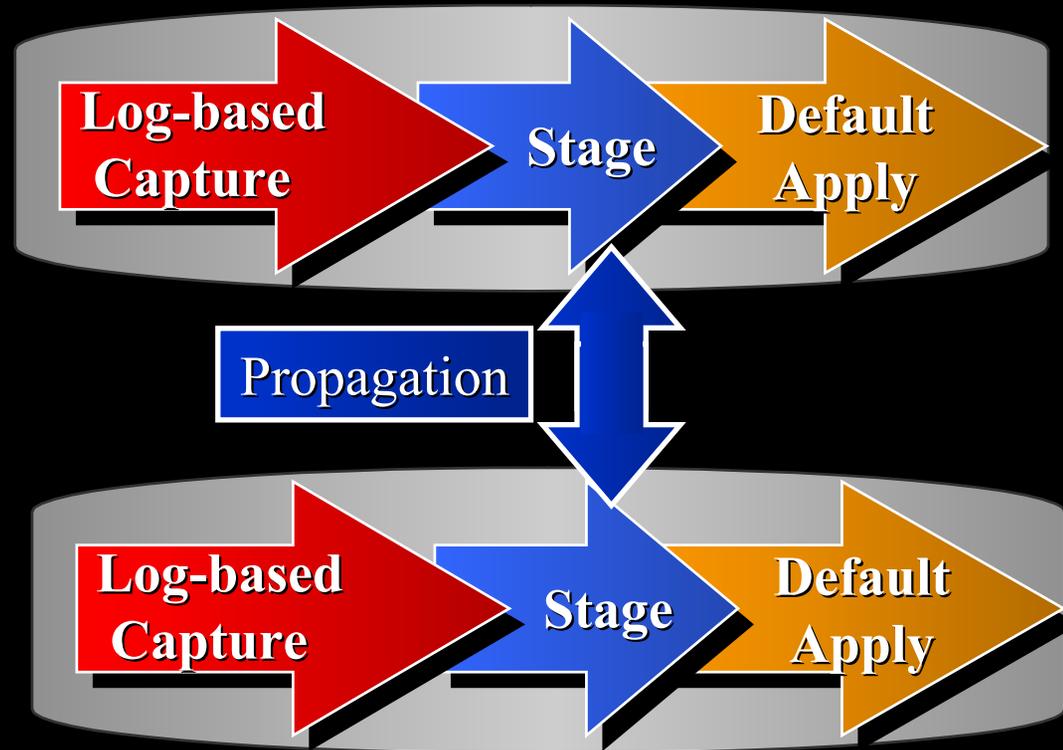
- Streams can be deployed to meet a variety of information sharing requirements
  - Replication
  - Data Warehouse Loading
  - Event Notification
  - Advanced Queuing

# Replication

- Streams asynchronously maintains multiple copies of objects via automatic apply
  - Identical objects
  - Related via a transformation or function
- Streams automatically captures, propagates, and applies DML and DDL changes
  - Detects and optionally resolves conflicts
- Supports flexible data movement and subsetting
- Gateways and APIs for heterogeneous support
- Compatible with Materialized Views

# Replication

- Benefits:
  - No quiesce for DDL
  - Lower overhead on production system
  - Reduced network traffic
  - Flexible configurations

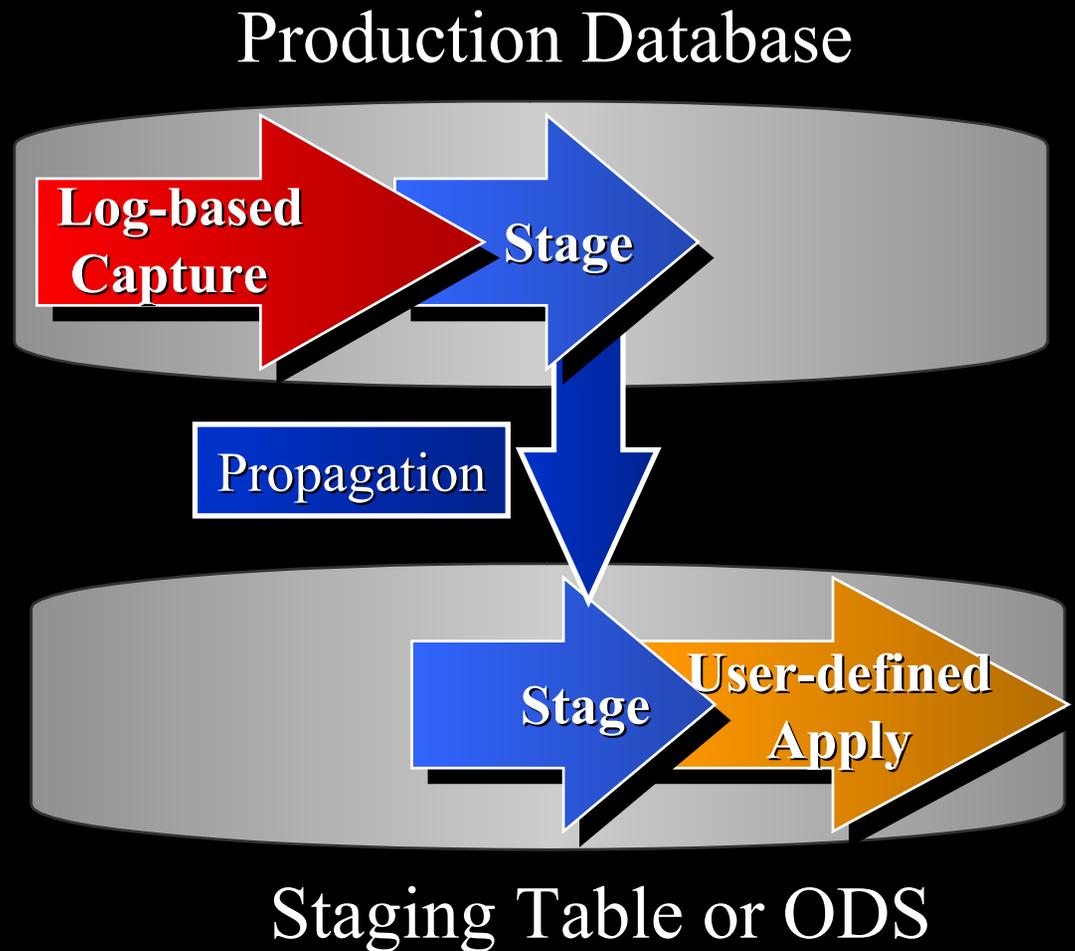


# Data Warehouse Loading

- Streams can load data warehouse staging areas and Operational Data Stores
  - Updates captured from a production system
  - Messages and business events from a process flow
- Supports continuous or batch loading
- Automatically transforms data to appropriate format and schema during Operational Data Store load

# Data Warehouse Loading

- Benefits:
  - low overhead
  - automatic transformation
  - near real-time loading of operation data stores

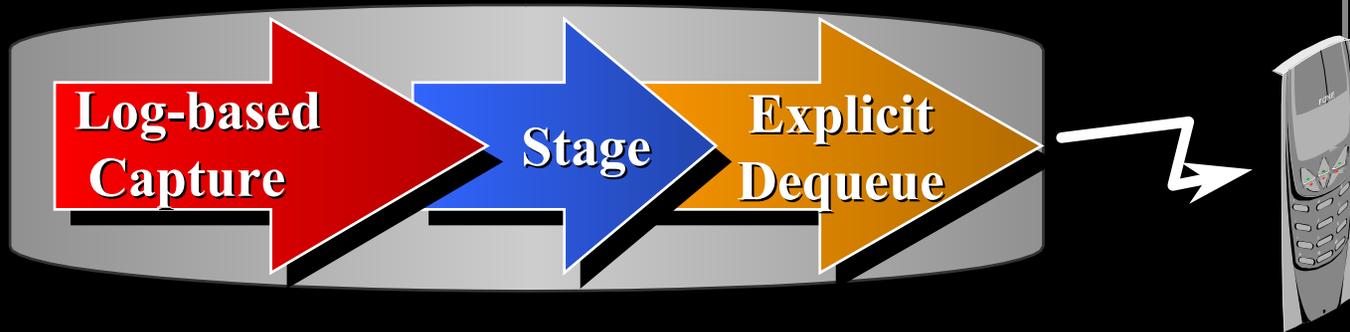


# Event Notification

- Streams can notify subscribers that events of interest have occurred
  - Pager notification of flight delays (Orbitz)
  - Notification of price drops (CNET Shopper)
  - Notification to sales manager of Gold Customer purchase (CRM App)
- Streams can evaluate DML events and send notifications to applications that send emails, page users, etc
  - Users get information they want

# Event Notification

- Benefits:
  - scalable
  - reduced custom development

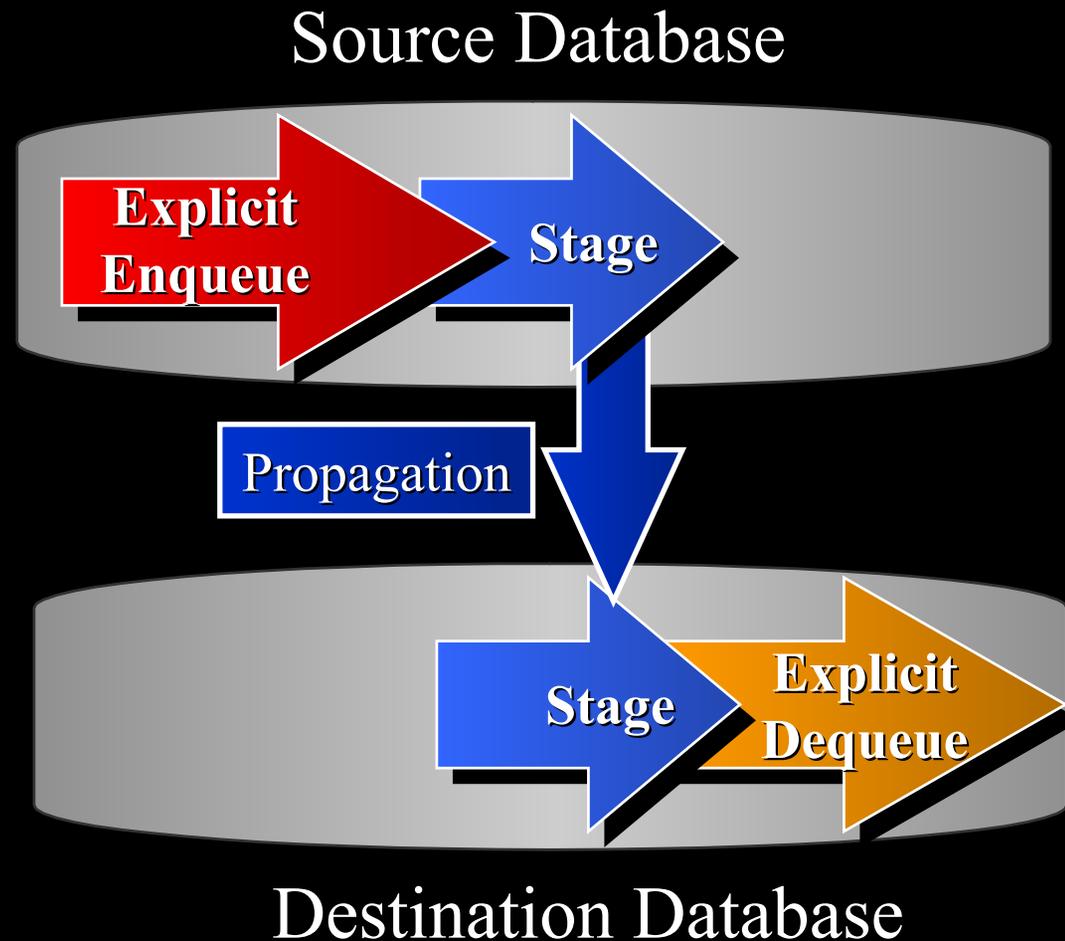


# Advanced Queuing

- Streams is the underlying infrastructure for Advanced Queuing, an enhanced database integrated message queuing solution
  - Point-to-point messaging, publish and subscribe
  - Single data, security and transactional model for database and message queuing operations
  - Centrally managed and multi-consumer queues to simplify configuration
  - Content-based subscriptions, internet access
  - Automatic dequeue to server-run user function
  - Automatic transform DML/DDL into messages

# Advanced Queuing

- Benefits:
  - Reduced development costs
  - Easy database integration
  - Single development, operational, security model
  - Reliability and integrity of database



# Other Oracle9i Information Sharing Features

- Advanced Replication Multi-master Replication
  - Provides compatible replication with Oracle 8, 8i, and 9i databases
  - Migration path to Oracle Streams in future release
- Materialized Views for Replication
  - Provides easy mass deployments of point-in-time copies of a database (or subset)
  - May be used in conjunction with Oracle Streams to deploy materialized view to remote servers

# Summary



- Oracle Streams unifies all enterprise information into a single Stream
  - Unifies database, messaging, replication, publish/subscribe APIs and capabilities
- Allows deployment of a variety of solutions
- Provides a single, unified solution to the problem of Information Sharing

A large, semi-transparent grey logo consisting of a stylized 'Q' and 'A' is centered in the background. A bright red ampersand (&) is overlaid on the 'Q' and 'A' in the center.

Q U E S T I O N S  
A N S W E R S

ORACLE®