# SAP
# C_HAMOD
## SAP Certified - Data Engineer - SAP HANA (C_HAMOD_2601)

**For More Information – Visit link below:**

**https://www.examsempire.com/**

**Product Version**

1. **Up to Date products, reliable and verified.**
2. **Questions and Answers in PDF Format.**



# Latest Version: 4.0
# Subjects

**1. Halosphere Aquapark: Group Attendance Analytics on SAP HANA Cloud**
**2. Dvina Tractor Works Export Sales Reporting on SAP HANA Cloud**
**3. Benguela Linen Services: HANA Cloud Reporting Model Setup**
**4. Akamas Carob Collective Stands Up Its First SAP HANA Cloud Analytics Layer**
**5. Rokel Mineral Sands: HANA Cloud Export Analytics Layer**
**6. Sermeq Glacier Water: Building an Export Analytics Layer**
**7. Kwanza Packaging: Unified Analytics on SAP HANA Cloud**
**8. Comalapa Ground Services: Ramp Analytics on SAP HANA Cloud**
**9. Bandama Palm Refiners Curated Analytics Stabilization**
**10. Cuscatlán Indigo Works Dye-Export Analytics Program**
**11. Zagros Saffron Collective Export-Season Analytics Build**
**12. Blue Mountain Fleet Leasing: HANA Cloud Utilization Analytics**
**13. Data Modeling with Calculation Views**
**14. Data Provisioning and Virtualization**
**15. Data Replication and Transformation**
**16. Security and Access in HANA Modeling**
**17. Model Performance Optimization**
**18. Modeling Environment and Project Lifecycle**

**Topic: 1**

**Halosphere Aquapark: Group Attendance Analytics on SAP HANA Cloud**

**Business Context**

Halosphere Aquapark operates three coastal water parks around Limassol, Cyprus, under one leisure brand. Each park runs its own gates, ride zones, and season-pass desk, but the commercial group is managed centrally. Leadership has asked for a single view of daily attendance across all three parks so the season can be planned from one place. Until now, each park reported its own numbers on its own spreadsheet, and the group had no shared figure that could be trusted for planning the coming summer.

**System Landscape**

The group runs its analytics on SAP HANA Cloud, and a data engineer builds the models in SAP Business Application Studio, working inside one modeling project. Each park records daily entry scans in its own table, and the three tables share the same column layout. Reseller partners also sell tickets online and send their sales records in each night. The engineer has started assembling a group attendance model, but so far the parks are still described only by their names and dates, with no combined figure that leadership can total.

**Reporting Context**

When the analytics lead opened an early build of the model, the reporting tool listed park names and calendar dates but offered nothing to sum for attendance. The three park tables were present, yet the group figure would not add up across them. A colleague suggested matching the park tables together on their scan columns, while another wanted only the largest park loaded first so the report would open quickly. Neither idea produced a group total that leadership could rely on.

**Data Onboarding**

The online resellers do not follow one convention. Their nightly ticket records arrive with channel codes written in mixed spellings and letter cases, so the same selling channel appears several different ways.

Visit us at: https://www.examsempire.com/c-hamod

The engineer wants those records standardized before they reach the reporting model, rather than leaving analysts to reconcile them by hand each morning. There is also a much larger body of past reseller records that a year-end report reads heavily, and that report is expected to keep running on nights when the reseller systems are offline.

**Publishing Context**

The engineer built a season-pass model in the project and told the analytics team it was ready, but the team cannot query it from the database. The model file is clearly present in the project, and it was even committed to the group's Git repository for safekeeping. Even so, the analysts' queries return nothing, and the season-pass figures the commercial team was promised have not appeared on any dashboard.

**Access Context**

Guest visit records carry a stored payment reference from the season-pass desk. Each resort manager should see only their own park's guest rows, and one manager was recently granted the privilege to open the guest-visit view. When the manager opened it, no guest rows came back at all. Leadership has also been clear that the stored payment reference must stay hidden from managers even where they are allowed to see their own park's guests, so the group's guest data is handled carefully as the shared model comes together.

**Your Assignment**

You are the data engineer for Halosphere Aquapark, working in a modeling project in SAP Business Application Studio over data in SAP HANA Cloud. You will assemble a group attendance model from three identically-structured park tables, provision and standardize reseller ticket records, publish a season-pass model for the analytics team, and scope guest data for resort managers. Work only with the modeling and data-provisioning tooling described; use neutral, supported settings rather than inventing exact identifiers or values

**1. CHALLENGE 1 — Unifying Per-Park Gate Scans into a Countable Attendance Measure**

• Combine the three identically-structured park gate-scan tables so their rows are stacked into one result set for the group.

• Model the combined view so that attendance is exposed as an aggregatable measure, not merely as an attribute.

• Where most reports open a single park, arrange the model so it skips the park sources a query's park filter does not need, without dropping any park from the model.

**CHECKPOINTS**

• The reporting tool can total attendance across all three parks.

• A single-park report does not scan the other parks' sources.

**2. CHALLENGE 2 — Standardizing and Provisioning Reseller Ticket Records**

• Standardize the mixed-case reseller channel codes as the records move into the model, rather than leaving analysts to reconcile them at read time.

• Choose a provisioning approach for a heavy year-end report so it can read a large body of past reseller records without depending on the reseller systems while it runs.

• Keep the sequence of the data flow correct, so each step follows the step it depends on.

**CHECKPOINTS**

• Cleaned reseller rows land in a target the reporting model reads.

• The year-end report runs when the reseller systems are offline.

**3. CHALLENGE 3 — Publishing the Season-Pass Model for the Analytics Team**

• Make the saved season-pass model consumable so the analytics team can query it from the database.

• Treat source-control versioning and database availability as separate steps, and confirm which one makes the model queryable.

**CHECKPOINTS**

• The analytics team can query the season-pass model from the database.

Visit us at: https://www.examsempire.com/c-hamod

• A Git commit alone is understood not to make the model consumable.

**4. CHALLENGE 4 — Scoping Guest Data to the Right Resort Managers**

• Ensure a manager can both invoke the guest-visit view and receive only their own park's guest rows.
• Obscure the stored payment reference for managers who are allowed to see their park's guest rows, without removing those rows or dropping the value for everyone.

**CHECKPOINTS**

• Each manager sees only their park's guest rows.
• The payment reference is obscured while the guest rows remain available.

## Question: 1

**CHALLENGE 1 — Unifying Per-Park Gate Scans into a Countable Attendance Measure**

The three coastal parks each record daily entry scans in a table with the same column layout, and the analytics lead wants one combined daily-attendance figure for the group.
Which modeling choice combines the three park tables so attendance can be totaled across all of them?

A. Combine the three park tables with a union node, so their rows are stacked into one result set that the model can then total across parks.
B. Join the three park tables on their gate-scan columns, so each park's rows are matched to the others before the attendance figure is totaled for the group.
C. Layer the three park tables under a rank node, so the highest-count park is kept for the group and its attendance figure is totaled for reporting.
D. Reference only the largest park's table in a projection node, so the attendance figure loads quickly and the two smaller parks are added at a later time.

**Answer: A**

Explanation:
The three tables share one column layout and simply need their rows stacked, which is exactly what a union node does; the combined rows can then be totaled as a group attendance measure. This keeps every park represented and produces the single trusted figure leadership asked for.

## Question: 2

**CHALLENGE 1 — Unifying Per-Park Gate Scans into a Countable Attendance Measure**

After the parks were combined, the reporting tool lists park and date attributes but offers no attendance figure to sum.
What is the underlying reason, and which correction fixes it?

A. The union node dropped the attendance column, so re-adding it as a restricted column in the same dimension view will let the reporting tool total it across the parks.
B. The park tables were combined on mismatched keys, so re-mapping those columns in the same dimension view will expose an attendance figure to total.
C. The view was built as a dimension calculation view, which exposes attributes but no aggregatable measure, so the model must be built as a cube calculation view.

Visit us at: https://www.examsempire.com/c-hamod

D. The reporting tool cached an empty result, so redeploying the same dimension view and clearing the tool's cache will make the attendance figure appear.

**Answer: C**

Explanation:
A dimension calculation view models attributes and has no aggregatable measure of its own, which is exactly why the tool offers nothing to sum. Rebuilding the model as a cube calculation view exposes attendance as a measure the tool can total.

## Question: 3

**CHALLENGE 1 — Unifying Per-Park Gate Scans into a Countable Attendance Measure**
Most reports open one park at a time, yet the combined view scans all three park sources on every query and is slow to open.
Which action reduces the scanning without dropping any park's data from the model?

A. Remove two of the park sources from the union node, so the view scans a single source while the other parks are reported from separate standalone park views.
B. Configure union pruning on the union node, so the view skips park sources that a query's park filter does not need while every park stays in the model.
C. Replace the union node with a join node, so only matching park rows are scanned and each single-park report returns a smaller combined result set.
D. Add a static cache over the whole combined view, so every park's figures are served from the stored copy even when only a single park is requested.

**Answer: B**

Explanation:
Union pruning lets the union node skip sources that a query's filter cannot match, so a single-park report reads only that park while all three parks remain modeled. It reduces scanning without changing the combined model.

## Question: 4

**CHALLENGE 2 — Standardizing and Provisioning Reseller Ticket Records**
Online resellers send ticket records with channel codes in mixed spellings and letter cases, and the team wants them standardized before the data reaches the reporting model.
Which approach standardizes the channel codes for the whole model rather than per report?

A. Apply a filter in the cube view that keeps only the already-recognized channel codes, so the mixed-case values are excluded from the attendance figures.
B. Add a restricted column in the cube view for each channel code, so the mixed values are grouped only when the reporting tool reads that one view.
C. Mask the channel-code column in the cube view, so the mixed-case values are obscured from analysts while the underlying reseller records themselves stay unchanged.

D. Build a flowgraph that reads the reseller records, standardizes the channel codes in a transformation node, and writes the cleaned rows to a target table.

**Answer: D**

Explanation:
A flowgraph transforms the data as it moves, so standardizing the channel codes in a transformation node and writing them to a target cleans the records once for every consumer. Analysts then read consistent codes without reconciling them each morning.

## Question: 5

**CHALLENGE 2 — Standardizing and Provisioning Reseller Ticket Records**
The engineer will build the reseller-standardization flowgraph.
Which ordering places each step after the step it depends on?

A. Add the transformation node, define the source reseller table, define the target table, then run the flowgraph.
B. Define the target table, run the flowgraph, define the source reseller table, then add the transformation node.
C. Define the source reseller table, add the transformation node, define the target table, then run the flowgraph.
D. Run the flowgraph, define the source reseller table, add the transformation node, then define the target table.

**Answer: C**

Explanation:
The source must exist before a transformation node can read it, the target must exist before the flow can write, and the flow can only run once its sources, node, and target are all defined. Option C follows that dependency order end to end.

## Question: 6

**CHALLENGE 2 — Standardizing and Provisioning Reseller Ticket Records**
A year-end report reads a very large body of past reseller records heavily and must keep running on nights when the reseller systems are offline.
Which provisioning approach fits the year-end report?

A. Replicate the reseller records into a table in SAP HANA Cloud, so the year-end report reads a local copy that does not depend on the reseller systems while it runs.
B. Create virtual tables over the reseller systems, so the year-end report reads the records live and always reaches back to the reseller systems every time that it runs.
C. Point the year-end report at the standardized target, so it reads the cleaned rows live from the reseller systems through the flow on every run.

D. Add a static cache over a virtual table, so the year-end report reads the reseller records quickly while it still depends on the reseller systems each run.

**Answer: A**

Explanation:
Replication stores a local physical copy in SAP HANA Cloud, so a heavy year-end report can read it independently of the reseller systems, including when they are offline. That independence and local read performance are exactly what the report needs.

## Question: 7

**CHALLENGE 3 — Publishing the Season-Pass Model for the Analytics Team**
The season-pass model is saved in the engineer's project and looks complete, but the analytics team cannot query it from the database.
Which action makes the model consumable for the analytics team?

A. Share the project folder with the analytics team, so they open the season-pass model in the editor and query it directly from there.
B. Export the season-pass model to a file, so the analytics team imports it into their own project and queries the copy they hold.
C. Grant the analytics team edit rights on the project, so they rebuild the season-pass model in their own space and query it there.
D. Deploy the season-pass model from the project, so the database holds a consumable object the analytics team can query.

**Answer: D**

Explanation:
A model saved in the project is not yet a database object; deploying it places a consumable object in the database that the analytics team can query. Deployment is the step that turns the design into something consumers can reach.

## Question: 8

**CHALLENGE 3 — Publishing the Season-Pass Model for the Analytics Team**
The engineer committed the season-pass model to Git and the analytics team still cannot query it.
What is the underlying reason the model remains unavailable?

A. The commit saved an older revision, so committing the current revision to Git will place the model in the database for the team to query.
B. A Git commit versions the project source but does not place an object in the database, so the model must still be deployed to be queried.
C. The commit targeted the wrong branch, so merging that branch in Git will place the model into the database for the analytics team to query later.

D. The commit excluded the model file, so adding it to the Git commit will place the model in the database for the team to query.

**Answer: B**

Explanation:
Git manages the project source code and its versions; it does not create a queryable database object. The model becomes consumable only when it is deployed, which is the step that was missing.

## Question: 9

**CHALLENGE 4 — Scoping Guest Data to the Right Resort Managers**
A resort manager was granted the privilege to open the guest-visit view but sees no guest rows, and each manager should see only their own park's rows.
Which action gives each manager their own park's rows?

A. Grant the manager a broader object privilege on the guest-visit view, so opening the view returns the guest rows for their own park.
B. Add the manager to the project as an editor of the guest-visit view, so their park's guest rows are returned whenever they open it.
C. Assign an analytic privilege that restricts the guest-visit view to the manager's own park, so only their park's guest rows are returned.
D. Add a fixed filter in the guest-visit view for the manager's park, so every manager who opens the shared view is limited to that park's rows.

**Answer: C**

Explanation:
Being able to open a view is an object-level right, but which rows a user actually sees is governed by an analytic privilege. Assigning an analytic privilege scoped to the manager's park returns exactly their rows and nothing else.

## Question: 10

**CHALLENGE 4 — Scoping Guest Data to the Right Resort Managers**
Managers who are allowed to see their park's guest rows must still not read the stored guest payment reference.
Which action hides the payment reference without removing those managers' rows?

A. Apply data masking to the payment-reference column, so its values are obscured while the manager still sees their own park's guest rows.
B. Remove the manager's analytic privilege on the view, so the payment reference is hidden along with all of that manager's park guest rows.
C. Drop the payment-reference column from the view, so no manager can read it and every guest row is served without that attribute at all.

D. Add an object privilege that blocks the payment-reference column, so the value is hidden while the manager keeps their own park's guest rows.

**Answer: A**

Explanation:
Data masking obscures the values of a specific column while leaving the rows in place, so managers keep their park's guest rows but cannot read the payment reference. It targets the column value without touching row authorization.

# Thank You for Trying Our Product

**Special 16 USD Discount Coupon: NSZUBG3X**

**Email:** support@examsempire.com

**Check our Customer Testimonials and ratings available on every product page.**

**Visit our website.**

**https://examsempire.com/**