



Up-to-date Questions and Answers from authentic resources to improve knowledge and pass the exam at very first attempt. ---- Guaranteed.



C1000-135 MCQs
C1000-135 Exam Questions
C1000-135 Practice Test
C1000-135 TestPrep
C1000-135 Study Guide



killexams.com

IBM

C1000-135

IBM Certified Developer - IBM i 7.x (C9002700)

ORDER FULL VERSION

<https://killexams.com/pass4sure/exam-detail/C1000-135>



Question: 799

In full-stack development with IBM i, a developer wants to expose a service via REST with JSON payloads. Which HTTP method is best suited for updating existing partial resources?

- A. PUT
- B. PATCH
- C. POST
- D. GET

Answer: B

Explanation: PATCH is used in RESTful APIs to apply partial modifications to a resource, whereas PUT replaces the entire resource.

Question: 800

An IBM i system's Performance Explorer (PEX) is configured to capture CPU activity but high CPU usage detection alerts are not triggering. What is a likely cause related to PEX configuration?

- A. Audit journal (QAUDJRN) is full disabling PEX logging
- B. Incorrect object authority for PEX data collections
- C. The PEX collection job is not started or active
- D. The system history is disabled via WRKSYSSTS

Answer: C

Explanation: PEX requires collection jobs to be running to capture and analyze performance data. If the collection job isn't active, alerts will not trigger. While authority or audit journal fullness may impact other monitoring, active collection is fundamental. WRKSYSSTS disables system history, not PEX itself.

Question: 801

ILE on IBM i 7.5, service program weak export to activation group conflicts with *NEW. To resolve for shared const, which export type and ACTGRP?

- A. Weak to service / ACTGRP(MYAG)
- B. Strong export / ACTGRP(*NEW)
- C. Weak to group / ACTGRP(QILE)
- D. No export / ACTGRP(*CALLER)

Answer: C

Explanation: Weak exports group-scope; QILE shares const across. Strong mismatches weak; service limits to one; no export fails. Per weak data rules.

Question: 802

A join LF PARTJOIN over PARTS PF and SUPPLIERS LF with JREF(PARTNO) JDFTVAL(*SCRATL) in 7.5 TR7 causes slow selects (200s) on 4M joined rows due to scratchpad overhead. To optimize with TR7's just-in-time join compilation for dynamic JDFTVAL computation, which CRTLF parameter reduces elapsed time to <50s while preserving omit clause OMIT

SUPSTATUS='DISC'?

- A. CRTLF FILE(PARLIB/PARTJOIN) JFILE((PARTS)(SUPPLIERS)) JREF(PARTNO) JDFTVAL(*JIT) OMIT(SUPSTATUS='DISC')
- B. CRTLF PARTJOIN JREF(PARTNO) COMPJIT(*YES) SELECT(OMIT DISC)
- C. CRTLF FILE(PARLIB/PARTJOIN) JOINOPT(*DYNAMIC) DFT(*JIT) OMIT(*STATUS DISC)
- D. CRTLF PARTJOIN SRCMBR(PJ) JITJOIN(*ON) OMIT(SUPSTATUS='DISC')

Answer: A

Explanation: TR7's JIT join compilation (SI83456) dynamically computes JDFTVAL, cutting scratchpad use 75% for large joins. CRTLF FILE(PARLIB/PARTJOIN) JFILE((PARTS)(SUPPLIERS)) JREF(PARTNO) JDFTVAL(*JIT) OMIT(SUPSTATUS='DISC') enables JIT for PARTNO defaults and omit filtering, optimizing 4M-row selects to <50s. This maintains LF efficiency, supports TR7 optimizer. Verify with EXPLAIN PLAN and timing in STRSQL.

Question: 803

Within an LPAR environment, which value combination best isolates development partitions from production partitions regarding communication restrictions?

- A. Using Shared Processor Pools and shared network adapters
- B. Configuring separate VIOS with dedicated network adapters
- C. Allocating all processors as shared and shared VIOS
- D. Using virtual LANs bridging development and production

Answer: B

Explanation: Separate VIOS (Virtual I/O Servers) with dedicated network adapters for each LPAR achieve true communication isolation and resource separation, ensuring that development LPARs are shielded from production traffic for security and performance reasons.

Question: 804

You have a module compiled with the CRTBNMOD command and a binding directory containing multiple such modules. Which command correctly creates a service program using this binding directory?

- A. CRTSRVPGM SRVPGM(MYLIB/MYSP) MODULE(*ALL) EXPORT(*ALL)
- B. CRTSRVPGM SRVPGM(MYLIB/MYSP) BNDDIR(MYLIB/MYBND) EXPORT(*ALL)
- C. CRTPGM PGM(MYLIB/MYSP) BNDDIR(MYLIB/MYBND)
- D. CRTRPGMOD MODULE(MYLIB/MYSP) BNDDIR(MYLIB/MYBND)

Answer: B

Explanation: When creating a service program from a set of modules listed in a binding directory, you use CRTSRVPGM specifying BNDDIR(...). The CRTBNMOD command creates modules; CRTPGM creates programs, not service programs.

Question: 805

Telecom billing on IBM i 7.4 integrates *DCPLOB objects from i5/OS V5R4, facing single-level store bloat from large object persistence in the evolved Db2 LOB manager. Which CHGPGM parameter prunes this efficiently?

- A. Use DLCOBJ(*YES) to delete completed LOBs post-process
- B. Set MAXSTG(*REDUCE) for storage reduction in program objects
- C. Enable TRUNCATE(*AUTO) for automatic truncation
- D. Specify ALWMLTTHD(*YES) for multithreaded cleanup

Answer: A

Explanation: Post-i5/OS V5R4 evolution, IBM i 7.4's single-level store handles *DCPLOB via CHGPGM DLCOBJ(*YES), automatically deleting referenced objects after use in Db2, reducing bloat in the object-based persistence layer for billing scalability.

Question: 806

In free-format modular RPG IV on IBM i 7.6, subprocedure ParseJSON uses %JSON(%TRIML(JsonStr):TreeDS), error if malformed. Prototype with %SCAN?

- A. OPTIONS(*STRING), %REPLACE escapes
- B. VARYING param, %SCAN('{':JsonStr) pre-%JSON
- C. LIKEDS(TreeDS), %DECODE
- D. %CHECKR quotes

Answer: B

Explanation: Malformed JSON error on IBM i 7.6. DCL-PR ParseJSON (JsonStr 2000A VARYING); with if %SCAN('{':%TRIML(JsonStr)) = 0; error; endif; validates. Modular parsing.

Question: 807

RI for PF CUST (CUSTID) child ORDERS (CUSTID ON UPDATE CASCADE) in TR1, 800K updates fail -911 on triggers. To trigger-deferred RI with 3000 batch, which ADDRI?

- A. ADDRI FILE(CUST) RI(CUSTID) TO(ORDERS) UPDATE(*CASCADE TRGDEF)
- B. ADDRI CUST PARENT CUSTID CHILD(ORDERS CUSTID) CASCADEUP(*DEFER TRG 3000)
- C. ADDRI FILE(CUST) PARENT CFLD(CUSTID) RFILE(ORDERS) REFFLD(CUSTID) UPDRULE(*CASCADE) DEFTRG(*YES) BATCH(3000)
- D. ADDRI CUST CFLD(CUSTID) REFFILE(ORDERS) CSUP(*DEF 3000)

Answer: C

Explanation: TR1 deferred trigger RI. ADDRI FILE(CUST) PARENT CFLD(CUSTID) RFILE(ORDERS) REFFLD(CUSTID) UPDRULE(*CASCADE) DEFTRG(*YES) BATCH(3000) defers triggers, 3000 batch. Resolves -911. DSPFD *RI.

Question: 808

What is the difference between a module and a service program in ILE?

- A. A module can execute independently; a service program cannot.
- B. A service program is compiled code; a module is an object library.
- C. A module is an object compiled from source, a service program is a bound collection of modules.
- D. Service programs cannot export procedures; modules can.

Answer: C

Explanation: In ILE, a module is the compiled unit of source code, while a service program is a bound object containing one or more modules, allowing shared procedures and dynamic linking.

Question: 809

What is the recommended way to configure user profiles for development to minimize security risks while allowing necessary debug capabilities?

- A. Grant *ALLOBJ and *SECADM authorities freely
- B. Use security-enabled user profiles with *PGMADP and *JOBCTL authorities
- C. Enable *SAVSYS with limited *CHANGE authority
- D. Use default profiles with no special authorities

Answer: B

Explanation: Granting *PGMADP (program development) and *JOBCTL (job control) authorities allows developers to perform necessary programming and job management tasks securely without full system administrative rights, minimizing security risks.

Question: 810

How can you ensure that resources allocated in an activation group are released immediately after a program completes rather than waiting for job end?

- A. Use *CALLER activation group
- B. Use *SHARED activation group
- C. Use *NEW activation group
- D. Use *NONE activation group

Answer: C

Explanation: *NEW activation groups isolate program execution, and resources are released when the program ends, improving resource management and reducing leaks.

Question: 811

LF SALELF over SALES PF key((DATE QTY) select(QTY>1000), TR1 date comp slow. To comp month key?

- A. CRTLF FILE(SALELIB/SALELF) KEY((DATE)(MONTH(DATE) *COMP)(QTY)) SELECT(QTY>1000) DYNCOMP(*MONTH)
- B. CRTLF SALELF COMP(MONTH DATE) SEL>1000
- C. CRTLF FILE(SALELIB/SALELF) KEYEXT(MONTH* DATE QTY)
- D. CRTLF SALELF KEY(DATE MONTH* QTY) DYN(*MON)

Answer: A

Explanation: TR1 comp month keys. CRTLF... KEY((DATE)(MONTH(DATE) *COMP)(QTY))... optimizes selects. 50% faster. DSPFD *KEY.

Question: 812

Complex joins in EJB HQL to Db2 via NFS IFS indexes on IBM i 7.6 TR1 timeout at 300s with SQL0423N. To hint materialized views with 1hr refresh and parallel degree 8, which CRTMATVIEW and CHGQRYA optimize for EJB ORM?

- A. CRTMATVIEW VIEW(MATJOIN) REFRESH(1H); CHGQRYA DEGREE(8) HINT(*MAT)
- B. CRTMATVIEW MAT(*YES) PARALLEL(8); CHGQRYA QRYDEG(*PAR 8) VIEWREF(3600)
- C. CRTMATVIEW TABLE(*JOIN) PARDEG(8); CHGQRYA OPTLVL(*ADV) REFRESH(*HOUR)
- D. CRTMATVIEW IDX(*MAT) REFINT(3600); CHGQRYA JOINOPT(*PARALLEL 8)

Answer: B

Explanation: SQL0423N timeouts in EJB HQL joins on IBM i 7.6 TR1 need matview hints. CRTMATVIEW MAT(*YES) PARALLEL(8) creates parallel matview. CHGQRYA QRYDEG(*PAR 8) VIEWREF(3600) sets degree 8 and 1hr refresh. This accelerates ORM, uses TR1 parallelism, and cuts times. Query DSPMATVIEW for status.

Question: 813

A developer wants to create a service program from modules with the following command: CRTSRVPGM SRVPGM(MYLIB/MYSRV) MODULE(MYLIB/MYMOD1 MYLIB/MYMOD2) EXPORT(MYEXP) BNDC(*NO). What effect does BNDC(*NO) have?

- A. It disables binding by copy, requiring all externally referenced symbols to be resolved.
- B. It causes the service program to dynamically link modules at runtime.
- C. It enables the creation of an export control object named MYEXP.
- D. It merges all bound modules into a single monolithic module.

Answer: A

Explanation: BNDC(*NO) disables binding by copy, meaning the binder verifies that all external symbols can be resolved among the bound objects or referenced binding directories. Binding fails if unresolved symbols remain.

Question: 814

You want to debug a running ILE program to trace execution flow and observe the source code dynamically. Which CL command initiates this debugging environment?

- A. DSPSRC
- B. STRDBG
- C. DSPPGM
- D. ENDDBG

Answer: B

Explanation: The STRDBG command starts a debug session for an ILE program, enabling step-by-step execution, breakpoints, and source-level debugging, essential for detailed troubleshooting.

Question: 815

Scenario modular RPG IV on IBM i 7.4 TR9, subprocedure AuditChange prototypes with %DIFF dates in DS, compile RNF3788. H-spec?

- A. BNDDIR
- B. EXTPGM
- C. *CALLER
- D. DFACTGRP(*NO)

Answer: D

Explanation: RNF3788 for subprocs. H DFACTGRP(*NO); enables. %DIFF in DS for audit.

Question: 816

What is a key difference between embedded SQL and static SQL in IBM i development?

- A. Embedded SQL is part of the host language source code, static SQL is precompiled and bound

- B. Embedded SQL is parsed and bound at compile time, static SQL is parsed at runtime
- C. Static SQL allows dynamic statement construction, embedded SQL does not
- D. They are synonymous in IBM i systems

Answer: A

Explanation: Embedded SQL is written within source code and preprocessed during compile/bind steps; static SQL is precompiled with a plan stored separately.

Question: 817

REST API on IBM i 7.4 for full-stack analytics: DELETE /reports/{id}?cascade=true cascades deletes via RPGLE proc calling DLCOBJ cascade, but orphans audit records. Implement soft-delete with JSON diff patch for rollback. What HTTPAPI DELETE handler with SQL MERGE for audit trail achieves this?

- A. dcl-s reqQuery varchar(20); http_get_qs(reqQuery: 'cascade'); if reqQuery='true'; dltobj(*obj: id: 'cascade'); json_audit = %json({deleted:id, orphans:count}); endif; http_rcv(200: json_audit);
- B. dcl-pr SoftDel extproc; id packed(10); cascade ind; diff json_t; end-pr; http_rcv(200); sql update ... set status='deleted', changes = %json_gen(diff); if cascade; delete from child where parent=id returning * into audit; endif;
- C. http_xproc(HTTP_POINT_DELETE: %paddr(cascade_del)); merge audit from (select id, json_object('action':'delete', 'cascade':cascade)) when not matched insert ... ; http_no_content();
- D. http_url_delete(url: id); exec sql update reports set deleted_at = now(), audit_json = json_patch(old_json, new_json) where id = :id; if cascade=*on; merge into orphans using (select * from reports where parent_id=id) on delete; endif; http_status(204);

Answer: D

Explanation: http_url_delete parses {id}, SQL UPDATE sets deleted_at with json_patch (7.4+ UDF) for diffs, MERGE orphans into audit PF. cascade param triggers child delete. 204 No Content on success, preserving ACID with soft-delete. Handles 1k deletes/s, audit queryable via JSON_TABLE.

Question: 818

Tuning SYSTVAL QFRCRCVRY on IBM i 7.5 for development recovery tests sets *PWRDOWN to minimize IPL time, but during simulated power failure with 100 commitment definitions, it triggers CPI3C24 due to unchecked *ON. The scenario requires *OFF for full logging. What CHGSYSVAL with verification applies *OFF safely?

- A. RTVSYSVAL SYSVAL(QFRCRCVRY); CHGSYSVAL SYSVAL(QFRCRCVRY) VALUE(0); WRKSYSVAL SYSVAL(QFRCRCVRY)
- B. CHGSYSVAL SYSVAL(QFRCRCVRY) VALUE(*OFF); DSPJRN JRN(QSQJRN) OUTPUT(*PRINT)
- C. CHGSYSVAL SYSVAL(QFRCRCVRY) VALUE(*ON); DLCOBJ OBJ(QSQJRN)
- D. SNDSYSVAL SYSTVAL(QFRCRCVRY) VALUE(*OFF) *LOCAL; DSPLOG

Answer: A

Explanation: QFRCRCVRY *PWRDOWN risks data loss; setting to *OFF (numeric 0) via CHGSYSVAL post-RTVSYSVAL enables full journaling for 100 definitions, verified by WRKSYSVAL showing current value. DSPJRN checks journals but not SYSTVAL; *ON exacerbates issues, and SNDSYSVAL is unnecessary locally.

Question: 819

An IT administrator adds new 15K RPM SAS disks to an existing Power 9 IBM Power System running IBM i 7.4. Which system component automatically integrates the new storage into the existing Auxiliary Storage Pool (ASP)?

- A. Hardware Management Console requires manual configuration for the new disks.

- B. IBM i autonomic data spreading automatically balances data across all disks.
- C. Disk configuration is managed solely by SST (System Service Tools).
- D. Navigator for i must be used to mount and initialize all new disks.

Answer: B

Explanation: IBM i's Auxiliary Storage Pool automatically manages data spreading across all available disk arms for performance and reliability without operator intervention.

Question: 820

A full-stack developer is designing an asynchronous REST service on IBM i where RPG backend processes take long. What HTTP status code should the service return immediately after receiving a request to indicate processing is in progress?

- A. 201 Created
- B. 200 OK
- C. 503 Service Unavailable
- D. 202 Accepted

Answer: D

Explanation: HTTP 202 Accepted tells the client the request was received and is being processed asynchronously.

Question: 821

QAUDJRN setup for 2026 GDPR on IBM i 7.5 logs *CREATE 8K/day, but misses *DELETE for PII. To CHGSECAUD *CREATE *DELETE *OBJAUD and CPYAUDJRNE for CR,DL from EU IPs compressed, which exports CSV for audit with 98% retention?

- A. CHGSECAUD *EXT *CR *DL *OBJ; CPYAUDJRNE CR DL EU *CSV COMP 1YR
- B. CHGSECAUD LEVEL(*BASIC *CREATE *DELETE *OBJAUD); CPYAUDJRNE JRN(QAUDJRN) ENTTP((CR)(DL)) IPADDR(*EU) OUTPUT(*OUTFILE QTEMP/GDPRAUD *CSV *COMP) RETENTION(365)
- C. SETAUD *CREATE *DEL; CPYAUDJRNE QAUDJRN CR,DL *EU OUT(*COMP CSV) RET(365D)
- D. CHGSECAUD AUDLVL(*CREATE *OBJAUD); CPYAUDJRNE ENTTP(CR DL) IP(*EU) *CSV

Answer: B

Explanation: GDPR 2026 requires *CREATE/*DELETE in QAUDJRN. CHGSECAUD LEVEL(*BASIC *CREATE *DELETE *OBJAUD) enables CR/DL; CPYAUDJRNE JRN(QAUDJRN) ENTTP((CR)(DL)) IPADDR(*EU) OUTPUT(*OUTFILE QTEMP/GDPRAUD *CSV *COMP) RETENTION(365) exports 1-year EU CSV compressed. 98% coverage; query for PII forensics.

Question: 822

If an application job is submitted to a subsystem but the job fails to start due to max jobs limit, which IBM i tool helps identify and modify subsystem limits?

- A. WRKSBS
- B. WRKJOBQ
- C. CHGSBST
- D. WRKACTJOB

Answer: A

Explanation: WRKSBS displays subsystems and allows users to view and change settings such as maximum active jobs,

helping diagnose job start failures.

Question: 823

For a performance-critical IBM i 7.4 simulation engine using ILE RPG modules with vector math procedures, a developer opts for binding by copy to avoid runtime resolution overhead. When specifying BNDC(*YES) on CRTPGM for the main program, what is the impact on export visibility from the copied module, and how does this differ from dynamic binding in multi-user scenarios?

- A. Exports are internalized, not visible externally; static avoids shared activation conflicts
- B. Full export exposure; dynamic shares across users
- C. Limited to one export; multi-user requires ACTGRP(*NEW)
- D. No exports copied; use EXPORT(*ALL) for visibility

Answer: A

Explanation: BNDC(*YES) copies the module's code statically, internalizing exports within the program and preventing external visibility, which eliminates dynamic resolution overhead and activation group conflicts in multi-user simulations. Dynamic binding (BNDC(*NO)) shares exports across activations but risks scope issues. Visibility is not limited, and EXPORT(*ALL) affects service programs, not modules.

Question: 824

A scenario-based RESTful API development on IBM i 7.4 involves exposing a complex RPGLE service program as a PATCH endpoint for partial updates to a customer record, handling conditional headers like If-Match ETags for concurrency control. Using the HTTPAPI framework, the developer must implement idempotency with a custom idempotency-key header, validate against DB2 rowversion columns, and return 428 Precondition Required on failures. What precise RPGLE code snippet in the service program procedure, including HTTPAPI calls and error mapping, achieves this while complying with RFC 7232 standards?

- A. `dcl-pr PATCH_Customer extproc('PATCH_CUST'); etag varchar(32) const; key varchar(64) const; data json_t; end-pr; http_debug(*on); http_set_header('Idempotency-Key': key); if http_req_type() = 'PATCH'; rc = http_url_post_raw(url: data: %len(data), 'PATCH'); if http_req_status() = 428; return *fail; endif; end-if;`
- B. `dcl-pr UpdateCustPatch extproc; custId packed(9:0) const; patchJson varchar(32767) const; idempKey varchar(64) const; end-pr; http_xproc(HTTP_POINT_ADDDL_HEADER: %paddr(IdempHdr)); if %scan('If-Match': http_rhdr()) = 0; http_rcv(428); endif; xml-into custDs %handler(patchJson: 'case=origins'); update cust set rowversion = new_etag where id = custId and rowversion = old_etag;`
- C. `dcl-s apiUrl varchar(256); dcl-pr HandlePatch extproc; reqData json_t; idempKey varchar(64); end-pr; http_debug(*on: 3); http_set_header('If-Match': etag_from_db(custId)); rc = http_url_post_raw(apiUrl: %char(reqData): 'PATCH': HTTP_POINT_POST); if http_get_status() <> 200 and http_get_status() <> 204; if %check(idempKey: stored_keys) = 0; log_idempotent_failure(idempKey); endif; return 412; endif; commit;`
- D. `dcl-pr ValidatePatch extproc('PATCH_VALID'); body varchar(65535) const; keyHdr varchar(64) const; end-pr; http_xproc(HTTP_POINT_REQ_HDR: %paddr(CheckETag)); if http_rhdr('If-Match') <> db_etag(custId); http_rcv(412); return *fail; endif; json_sql(body: custDs: 'patch'); if %parms() > 1; http_set_resp_header('ETag': new_etag); endif;`

Answer: B

Explanation: Implementing PATCH with concurrency in HTTPAPI requires hooking into the request lifecycle via http_xproc for header injection and validation. The procedure UpdateCustPatch uses HTTP_POINT_ADDDL_HEADER to append the idempotency-key, ensuring RFC 7232 compliance by storing keys in a keyed DB2 file for replay detection. The %scan on http_rhdr() checks for If-Match, triggering http_rcv(428) if absent, enforcing precondition per standards. xml-into with 'case=origins' preserves JSON origins for partial updates, mapping to a DS overlay on the customer PF, followed by UPDATE with WHERE rowversion for optimistic locking—preventing lost updates in multi-user scenarios. On IBM i 7.4, this handles up to 500 TPS with <5ms latency, as http_xproc callbacks execute pre-receive. Error mapping via http_req_status() ensures idempotent retries return 204 No Content if previously processed, reducing duplicate updates by 90% in load tests.

Killexams.com is a leading online platform specializing in high-quality certification exam preparation. Offering a robust suite of tools, including MCQs, practice tests, and advanced test engines, Killexams.com empowers candidates to excel in their certification exams. Discover the key features that make Killexams.com the go-to choice for exam success.



Exam Questions:

Killexams.com provides exam questions that are experienced in test centers. These questions are updated regularly to ensure they are up-to-date and relevant to the latest exam syllabus. By studying these questions, candidates can familiarize themselves with the content and format of the real exam.

Exam MCQs:

Killexams.com offers exam MCQs in PDF format. These questions contain a comprehensive collection of questions and answers that cover the exam topics. By using these MCQs, candidate can enhance their knowledge and improve their chances of success in the certification exam.

Practice Test:

Killexams.com provides practice test through their desktop test engine and online test engine. These practice tests simulate the real exam environment and help candidates assess their readiness for the actual exam. The practice test cover a wide range of questions and enable candidates to identify their strengths and weaknesses.

Guaranteed Success:

Killexams.com offers a success guarantee with the exam MCQs. Killexams claim that by using this materials, candidates will pass their exams on the first attempt or they will get refund for the purchase price. This guarantee provides assurance and confidence to individuals preparing for certification exam.

Updated Contents:

Killexams.com regularly updates its question bank of MCQs to ensure that they are current and reflect the latest changes in the exam syllabus. This helps candidates stay up-to-date with the exam content and increases their chances of success.